METHOD AND APPARATUS FOR CONDUCTING DYNAMIC VOLUME TRACKING IN MOUNTABLE FILE SYSTEM

Publication number: JP3171238 (A)
Publication date: 1991-07-24

Inventor(s): BURAIAN EMU UIRUMAN; MAAKU JIEI ZUBIKOOSUKI;

JIEIMUZU JII RETSUUIN; RAJIEN JIYAYANTEIRARU SHIYAA

4

Applicant(s): MICROSOFT CORP +

Classification:

- international: G06F12/00; G06F15/16; G06F15/177; G06F17/30; G06F12/00;

G06F15/16; G06F17/30; (IPC1-7): G06F12/00; G06F15/16

more >>

Also published as:

JP7104808 (B)

JP2090000 (C)

EP0415346 (A2)

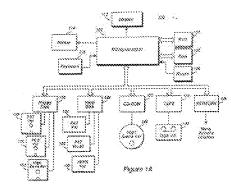
EP0415346 (B1)

- European: G06F17/30F

Application number: JP19900227905 19900829 Priority number(s): US19890400531 19890829

Abstract of JP 3171238 (A)

PURPOSE: To automatically map a file system in an indeterminate medium by comparing a volume identifier read from a medium with a identifier related to the file system. CONSTITUTION: The volume identifier is read from the medium by a loaded file system driver to be compared with the identifier related to the file system driver. Then when this identifier and the volume identifier coincide with each other, the file system driver is mounted. On the other hand when both identifiers do not coincide with each other, a next file system identified by the linked list of the file system driver is loaded. Thereby regardless of the form and the format of the file system the indeterminate medium is automatically and dynamically mapped to a proper file system.



Data supplied from the espacenet database — Worldwide

⑩日本図特許庁(JP)

① 特許出題公開

◎ 公 開 特 許 公 報 (A) ₩3--171238

@Int. CL.*

識別記号

庁内祭理番号

每公開 平成3年(1991)7月24日

G 08 F 12/00 15/18 3 0 1 370 M 8944-5B 6945-5B

審査請求 未請求 請求項の数 1 (全29頁)

斜発明の名称

設置可能なファイルシステムにおいてダイナミックボリュームトラ ツキングを行う方法及び装置

> M 72-227905 (2)特

@# 願 平2(1990)8月29日

優先權主張

70条明 者

アメリカ合衆国 ワシントン州 98007 ペルヴィュー

ブライアン エム ウ

1308 エヌ ノースイースト フォーティサード ブレイ

ス 14545

(D) 類人 マイクロソフト・コー

ボレーション

イルマン

アメリカ合衆国 ワシントン州 98052~6399 レッドモ

ンド ワン マイクロソフト ウエイ(番地なし)

90代 理 人

弁理士 中 村 稔 外 7 名

最終質に続く

215 183

し発明の名称

数置可能なファイルシステムにおいてダイナ ミックポリュームトラッキングを行う方法及び ** **

2.特許請求の範囲

データ配憶装置とコンピュータシステムとの 通信に使用するファイルシステムをマウントす る方法において、

- (4) 前記ファイルシステムがリングされたジ ーケンスに緩放されている省略時ファイルシス テムを備えたコンピュータオペレーティングシ ステムに複数のファイルシステムモジュールを 数分、
- b) 前記コンピュータシステムにデータ記憶 装置を接続し、
- 5) 前記コンピュータシステムが簡記データ 記憶装置にアクセスする最初の時点で、前配デ 一夕記憶装置におけるメディアの変化を検出し、
 - 4) 前紀ファイルシステムのリストにおいて

識別されたファイルシステムをローディングし、

- e) 煎貂メディアから、ローディングされた 前記ファイルシステムにより位置が特定化され たポリューム識別子を読み取り、
- (1) 前記メディアから読み取られた前記ポリ ューム機獅子を、前記ファイルシステムに関連 する機別子と比較し、
- g) 前記両機別子が一致する場合には、前記 ファイルシステムをマカントし、
- h) 前記調識別子が一致しない場合には、フ ティルシステムの前続リストにおいて識別され た次のファイルシステムをローディングし、
- コトファイルシステムの前続リストにおける 各ファイルシステムが試験されるまで又は一致 が見出されるまで、前記工程(e)に戻り、
- 1) 一致が全く見出されない場合には省略時 ファイルシステムをマウントすることを特徴と するデータ記憶装置とコンピュータシステムと の遺信に使用するファイルシステムをマウント する方法。

3.発明の詳細な展明

本明報書には、385個のフレームを含むマイ クロフィッシェの4枚のシートからなる付録「 (appendixi) が含まれている。

本発明はコンピュータ制御システムの分野に関 し、より詳しくは、コンピュータシステムを備えた 装置領土の適信を行う方法及び手段に関する。

一般に、コンピュータシステムは、中央処理装置と、ラングムアクセスメモリと、リードオンリメモリと、データ人力装置、データ出力装置、フロッピディスク及び固定ディスク又はハードディスク等の種々の不揮発性データ配信装置等の種々の周辺装置とを有している。一般に、それぞれの装置間の遺信はコンピュータオペレーティングシステムにより制備される。良く知られた1つのコンピュータオペレーティングシステムとして、マイクロソフト社(Microsoft)から市販されているMS-005オペレーティングシステムがある。

MS、DOSオペレーティングシステムにおいては、 単一のファイルシステムが、周辺装置に配憶され

ice dependant drivers)との間の遺伝を容易に行うことができる。また、ファイルシステムは、オペレーティングシステムのカーネルにより発せられた欲取り及び者込みコマンド(並びに、ファイルを開閉する機能)を、デバイスドライバが認識できるフォームに変換することに応答することができる。

18-805 オペレーティングシステムを用いる場合には、オペレーティングシステムは、コンピュータシステムに用いられている特定の周辺装置に使用できる適合ファイルシステムを構成しなければならない。一旦ファイルシステムが構成されたならば、このファイルシステムは、オペレーティングシステムが変更されることはない。このため、一般に、広範囲のプログラム作成努力と多大の消費によが要求される。また、コンピュータオペレーティングシステムの詳細にアクセスできない人は、ファイルシステムを容易に変

たファイルの編成を記載しかつ構成している。コ ンピュータシステム及びそれぞれの周辺装置の両 者により認識されたフォーマット中のデータをコ ンピュータシステムが終み取り又は概念込みでき るようにするには、データはこのファイルシステ よに従って鑼放されなくてはならない。 顔えば、 MS-808オペレーティングシステムに使用される従 来のフロッピディスクを削いた周辺装置において は、フロッピディスクのデータは、FAT ファイル システム (ファイル割当てテーブル (lile allacation table) を用いていることから、このよう に命名されている)として知られているファイル システムに従って構成されている。PAT ファイル システムは、今日、世界中で最も広範囲に使用さ れているファイルシステムの1つである。テーブ 紀復装置のような、周辺装置の他の形式のデータ 記憶装置には、他のファイルシステムを接続する こともできる。

ファイルシステムにより、オペレーティングシ ステムのカーネルとデバイス提携ドライバ(day-

更することはできない。

また、従来のシステムにおいては、路種ファイ ルシステム (foreign file systems) のファイル を収容しているディスクメディアを、固有システ ム (mative system)に使用することはできない。 例えば、多数の製造業者(各製造業者は別々のフ ァイルシステム構成に準拠している)により多く のコンピュータシステムが多年に至って開発され ている。現在のスタティックファイルシステムの 技術では、一般に、或るしつのシステムからのデ ィスクメディアを別の形式のシステムで機能させ ることはできない。コンピュータは一層ポピュラ 一なものとなっているため、あらゆる形式のコン ビュータシステムの間でファイルを共用できるよ うにすることの重要性が増大している。事実上知 られているあらゆるコンピュータシステムからの ディスクメディアを、単一のオペレーティング際 境において自動的に認識しかつ読み取ることがで きるシステムは未だ存在しない。また、コンピュ --タオペレーティングシステムのカーネルを変え る必要なくして、成るシステムに付加(又は変更) できるファイルシステムは来だ存在しない。

簡単に云えば、本発明は、コンピュータシステ ムに使用されるメディアを自動的に識別しかつ前 紀メディアを認識するファイルシステムを自動的 かつダイナミックにマウントできる方法及び手段 に関するものである。本発明の好ましい実施例に よれば、省略時ファイルシステム (default file system)を備えていて、リンクされたシーケンス に従ってファイルシステムを総成するように構成 されたコンピュータシステムに、1つ以上のデー 夕紀憶装置及び複数のファイルシステムドライバ を設けることができる。このコンピュータシステ ムは、終コンピュータシステムの全ての閉辺装置 を連続的にモニタリングしていて、周辺記憶装置 におけるメディアのあらゆる変化を検出するよう になっている。データ配復装置のメディアが変更 されるあらゆる場合、又は最初にコンピュータン ステムがデータ記憶装置にアクセスする場合には、 ファイルシステムドライバのリストにおいて識別

が、多数のメディア形式の全てを識別できるよう にし、かつ適正なファイルシステムをマウントし て前紀メディアに使用できるようにする方法及び 手段を提供することにある。

本発明の他の目的は、不確実メディアにファイルシステムを自動的にマッピング (配置) する方法及び手段を提供することにある。

本発明の他の目的は、ユーザからのインタラクションなくして、自動的に不確実メディアに適用 できるコンピュータンステムを提供することにある。

本発明の他の目的は、オペレーティングシステムのカーネルを変更する必要なくして、ファイルシステムを変更でき又はコンピュータオペレーティングシステムに付加できるようにコンピュータオペレーティングシステムを改善することにある。

本発明の他の目的は、メディアのフォーマット のあらゆる従属性が適当なファイルシステム内に 密閉 (encapsulated) されている不確実なメディ アにファイルシステムを自動的にマッピングでき された最初のファイルシステムドライバがローディングされ、ローディングされたファイルシステムドライバによりボリューム機別子(volume identifier)の位置が特定化されているメディアからボリューム機別子が読み取られる。メディアから読み取られたこのボリューム機別子は、次に、ファイルシステムドライバに関連している機別子と比較され、この機別子とボリューム機別子とがロー致するときにはファイルシステムドライバがマウントされる。両機別子が一致しない場合には、ファイルシステムドライバのリンクされたリストにおいて機別された次のファイルシステムドライバがローディングされる。

その後、このプロセスは、ファイルシステムドライバのリンクされたリストにおける各ファイルシステムドライバが試験されるまで、又は一致が見出されるまで繰り返される。一致が見出されない場合には、省略時ファイルシステムがマウントされる。

従って本発明の目的は、コンピュータンステム

る方法及び手段を提供することにある。

本発明の更に他の目的は、任意に設置できるファイルシステムをコンピュータシステムに設けることかできるようにする方法及び手段を提供することにある。

本発明のこれらの目的及び他の目的は、旅付図 面を参照して以下に述べる本発明の詳細な説明に より明らかになるであろう。

第1図には、本発明の原理に従って構成されたコンピュータシステム100が示されている。このコンピュータシステム100は、中央処理装置すなわちマイクロブロセッサ102と、ランダムアクセスメモリ104と、リードオンリメモリ106と、マウス108及びキーボード110のような入力装置と、ディスプレイ112及びブリンタ114のような出力装置と、フロッピディスクドライブ116、ハードディスクドライブ120。CB-BONドライブ122放びテーブドライブ124等からなる種々の不揮発性記憶装置とを有している。また、このコンピュータシステム100は、

ネットワーク 1 3 6 と適信できるようになっている。不揮発性記憶とは、装置の電源を遮断しても データが消去されないことをいう。

従来のシステムにおいては、オペレーティング システムは、各周辺装置が単一のメディア形ファ イルシステムドライバのみと互換性をもつファイ ルシステムドライバによりスクティックに構成さ れている。指定のファイルシステムドライブとの 互機性のないドライブにメディアが供給されると、 メディアは首尾良くアクセスすることができない。 以下に説明するように、本発明は、周辺装置とは 独立してかつメディアに関するデータのフォーマ ット又は位置についての条件を緊踝することなく して、関連するファイルシステムにメディアを自 動的にマッピングする方法及び手段を提供するも のである。例えば、フロッピドライブユニット (プロッピディスクドライブ) 116は、多数の ファイルシステムに従ってフォーマット化された ボリューム(例えば、PAT ファイルシステムに従 ってフォーマット化されたポリューム128、段

テムにより調整することができる。しかしながら、 本発明は特に、Microsoft 社により開発されたOS /2ォペレーティングシステムに使用するのに適し ている。本発明の作動環境 (operating environmeni) の構成が第18図に示してある。一般に、 アプリケーション152は、カーネル154によ り処理されるファイルシステムのリクエストを発 生する。次いで、カーネル154は、このリクエ ストを適当なファイルシステムドライバ (850) 158~170に薄く、任意のファイルシステム ドライバを、多数のハードウェア装置と協働させ ることができる。例えば、ポリューム17%、174 についてファイルシステム作動を行う場合には、 Bigh Sierra ファイルシステム 1 5 6 をCB-808ブ レーヤ(CB-80M ドライブ) 122及びディスクド ライブ!! 6に使用することができる。同様に、 PAT ファイルシステム1 6 0 及びBPFSファイルシ ステム162の顕著は、ボリェーム176、178 (これらの各々は、ハードディスク 120にある) についてのファイルシステム作動を行うのに使用

く知られたHigh Sierra ファイルシステムに従っ てフォーマット化されたボリューム132、及び もうしつのファイルシステムに従ってフォーマッ ト化されたポリューム130)に使用することが できる。同様に、ハードディスク(ハードディス カドライブ〉120の種々の区分(バーティシャ ソ)は、ボリューム134、136、138とし て表した多数のファイルシステムに従ってフォー マット化することができる。阿様に、CO-808ドラ イブ122及びテーブシステム (テーブドライブ) 1244, #12-4140, 142 (2464, それぞれのファイルシステムに従ってフォーマッ ト化されている)に使用することができる。また、 ネットワーク128は、サーバ(該サーバは、そ れる自体のファイルシステムに従って作動する) を構えた任意の数のネットワークに接続すること ができる。

コンピュータシステム 1000作動(オペレー ション)は、良く知られた多数のオペレーティン グシステムのうちの任意のオペレーティングシス

することができる。また、ポリューム180についてファイルシステム作動を行う場合には、ディスクドライブ116にファイルシステムドライバを使用することができる。従って、本発明によれば、ファイルシステムの形式及びフォーマットの知何に係わらず、適当なファイルシステムに不確実メディアを自動的にかつダイナミックにマッピングする方法及び手段が提供される。

類 2 A 図は、従来技術による RS-BOS オペレーティングシステムのファイルシステム構成を示すものである。 RS-BOS オペレーティングシステム200 においては、オペレーティングシステム 2 0 2 が埋設されている。この FAT ファイルシステム 2 0 2 はオペレーティングシステムのカーネル 2 0 4 内に一体化されているため、変更 (モディファイ) することは困難である。また、付加的なファイルシステムが必要な場合には、オペレーティングシステムのカーネル 2 0 4 を割き替えてそれらのファイルシステムに適合できるようにしなければなら

ない。

本発明によれば、第28回に示すシステム技術により上記問題点を解決することができる。本発明のコンピュータシステム100においても、05/2カーネル252内には、PATファイルシステム202が理殺されている。しかしながら、本発明によれば、オペレーティングシステムのカーネル252に対して外部装置であるFATファイルシステムドライバ254、256、258をダイナミックに取り付ける方法及び手段が提供される。随前には、設置可能な3つのファイルシステムドライバの数に観光たシステム250が売されているが、実際には、本発明は、ファイルシステムドライバの数に制策されることはない。

設置可能 (installable) なファイルシステム ドライバ (file system driver。「FSB」) は、 多くの点でデバイスドライバに類似している。PSD は、ダイナミックリンクライブラリ (dynamiclink library、「DLU」、一般には、SYS 又はIFS エクステンションを猶えている))のように構成さ

う。カーネルは、セクタ1/8 に対するデマンドを 適当なデバイスドライバに導き、かつその結果を FSD に関す。

ボリュームをFSBs (複数のFSB)と結合させるべくオペレーティングシステムにより用いられる手 類は、ダイナミックボリュームマウンティング (dynsmic volume mounting) と呼ばれ、次のように作動する。ボリュームが最初にアクセスされるとき、戦いは、直接アクセスを行うべく (例えば FORMATオペレーションにより) ボリュームがロックされた後に、オペレーティングシステムのカーネルは、ボリュームから FSBsの各々への識別情報を発生し、これは、FSBがこの情報を認識するまで環次行われる、FSBが ボリュームをクレイムすると、ボリュームがマウントされ、ボリュームに対する全ての連続ファイル1/0 リクエストが、ボリュームをクレイムした FSB に遅かれる。

この構成により、従来技術にはない幾つかの利 点を得ることができる。例えば、不確実メディア

れたファイルのディスク上に存在し、COMFIG.SFS ファイルにおけるIPS エステートメント(statesents) によるシステムの額期化中にローディング される。IPS ×置置(directives)は、それらが出 合う命令において処理され、また、デバイスドラ イバについてのDEVICE - 文(statements)の命令に 対して感応する。これにより、ユーザが、非機維 デバイス用のデバイスドライバをローディングし、 綾デバイスのボリュームからファイルシステムド ライバをローディングすること等が可能になる。 一旦850 が設置されかつ初期化されると、カーネ ルは、ファイルの開放、競取り、書込み、シーク (seeks) 、開鎖等についての論理的リクエスト (logical request)の言語で、FSO と運信する。 PSD は、ボリュームそれ自体に見出される額額積 成及びチーブルを用いて、これらのリクエストを セクタ競取り(sector reads)用のリクエストに翻 鉄も、かつ、ファイルシステムベルバ(File Syl stex Helpers、「Fallies」) と呼ばれる特別なカ ーネル人口点を呼び出すことができる数込みを行

がコンピュータシステムに与えられる場合に、コンピュータシステムは、利用できるファイルシステムは、利用できるファイルシステムドライバを位置付けすることができ、これにより、メディアへのファイルシステムドライバの自動マッピングを行うことが可能になる。また、オペレーティングシステムのカーネルを変更する必要なくして、ファイルシステムドライバを更新することができる。更に、新しい形式の周辺装置が開発されたときに、既存システムのノフトウェアを混乱させることなく、適当なファイルシステムドライバをオペレーティングシステムに付加することができる。

コンピュータンステム 100のより詳細なダイ アグラムが第3回に示してある。コンピュータン ステム 100は、アブリケーションプログラム 302とディスク装置304のようなデータ記憶 装置との間の通信を行うことができるオペレーティングシステムのカーネル252を有している。 また、このコンピュータンステム 100は、ファ イルシステムドライバを54~258と関連して作動するデバイスドライバる06を有している。 図面には単一の周辺装置304を鍛えたコンピュータシステム100が示されているが、本発明は、任意の数の論理的又は物理的周辺装置と組み合わせて使用できるものである。

作動に際し、アプリケーションプログラム302 は、所望の機能についての人口点を呼び出すこと により、オペレーティングシステムのカーネル 252に対する論理的ファイルリクエストを発行 する。これらの機能には、ファイルを開放するこ と (Bos Open)、ファイルを読み取ること (Bos Read)、ファイルに書き込むこと (Bos Write) 等の リクエストを含めることができる・オペレーティ ングシステムのカーネル252は、これらのリク エストを、ファイルを保持(ホールディング)す る特定のポリュームについての適当なファイルシ ステムドライバ254~258に導く。次に、適 当な設置可能なファイルシステムドライバが、論 理的ファイルリクエストを、指定メディアの論理

えば、コンピュータシステムが最初にブート(60ot) されるとき、ポリュームが最初にアクセスさ れるとき、又はコンピュータシステムが、ディス ク装置304内に不確実メディアが存在すること を決定するときにはいつでも、コンピュータシス テムは、ファイルシステムドライバのリンクされ たリストにおける機材のファイルシステムドライ バを試験する。ファイルシステムドライバがディ スク装置にローディングされたポリュームを認識 する場合には、ファイルシステムドライバがマウ ントされる。そうでない場合には、コンピュータ システムは、メディアを認識するファイルシステ ムドライバが位置付けされるまで、利用できるフ ァイルシステムドライバを連続的にポーリングす る。関心をもつメディアを認識する設置可能なフ ティルシステムドライバが全く晃出されない場合 には、省略特ファイルシステムドライバがマウン トされる。本発明の好ましい実施機においては、 省略時ファイルシステムは、上記のFAT ファイル システムである。

的セクタの換取り及び審込みのためのリクエストに翻訳し、かつオペレーティングシステムのカー まルのファイルシステムへルバ308を呼び出し て、これらのリクエストを適当なデバイスドライ バ306に薄く。ファイルシステムへルバ(Fallipa)308については、以下により詳細に説明 する。デバイスドライバ306は、オペレーティ ングシステムのカーネルからの論理的セクタリク エストを、特定の物理ユニット(すなわち、メディアのシリンダ、ヘッド及びセクタ)についての リクエストに変形し、かつ、ディスク装置にコマ ンドを発行して、ディスクメディアとランダムア クセスメモリ310との間にデータを伝達する。

次に、物理装置を特定のファイルシステムにマッピングすることについて以下に詳細に疑問する。
MS-BOS環境(MS-BOS environment)においては、フロッピディスクはボリュームと呼ばれる。固定ディスク (又はハードディスク) は、多数のボリュームに気分することができる。このターミノロジが、本発明に普尾良く適用されている。簡単に云

不確実メディアは、幾つかの方法により検出す ることができる。ディスク装置には機械的なラッ チ機構が設けられており、該ラッチ機構は、ディ スクがディスク装置から取り出されるとき又はデ ィスク装置に装填されるときに作動する。一般に ラッチ機構は、ドライブの次の作動によりドアが 欝放されたことを示すように機能する。デバイス ドライバがこの裏景を受け取ると、エラー不確実 メディア (ERROR UNCERTAIN MEDIA)がオペレーテ ィングシステムに戻される。機械的なラッチ機構 がないシステムにおいては、所定時間より短い時 間内にメディアを変更できないと考えられる。本 発明の好ましい実施例においては、この時間は2 砂であると考えられる。従って、所定時間以上の 時間をかけても特定のボリュームがアクセスされ ない場合には、このメディアは不確実であると推 定される.

第4回は、PAT ファイルシステムのディスクフォーマットのダイアグラムである。PAT ファイルシステムは、MS-86Sオペレーティングシステムの

初期から該85-B05オペレーティングシステムに使 知されている。FAT ファイルシステムについての 詳細な説明が、Ouncan署、「アドバンスMS BOS プログラミング("Advance BS SSS Programming")] (Microsoft Press 社刊、1986、1988) において なされている。FAT ファイルシステムは、FAT フ ァイルシステムは、ファイル割当てテーブル (ドー He Allocation Table) を中心題目としている。 各論理的ポリュームはそれ自体のFAT と関連して いて、2つの重要な機能を有している。すなわち、 客論理的ポリュームは、割当てユニット (allocetion snits)のリンクされたリストの形態をなす ポリュームに関する各ファイルについての割当て 情報 (allocation information) を収容していて、 その割当てユニットには、創出(又は拡大)され ているファイルへの創造で (代入、asaisment) がないことを表示する。

FAT ファイルシステムに従ってディスクがフォーマット化されると、ブートセクタ (boot sector) がセクタゼロと書き込まれる。ファイル翻当

の使用が記述される。ディレクトリの各ファイルの人口には、これらのファイル(該ファイルは、FATへの人口点として使用される)に割り当てられる優初のクラスタの数が収容される。人口点から、各PATスロット(PAT slot)は、機終クラスタマークに出合うまで、ファイル内の次のクラスタの番号を収容する。また、PATファイルシステムには、読取りエラー等によるPATのセクタへのアクセスが失敗した場合に用いることができる機初のファイル割当てテーブルの複類を維持する機能をオブションとして設けることができる。

ファイル額当てテーブルの後には、ルートディ レクトリが続く。このルートディレクトリは、ファイル、他のディレクトリ、及びオブションとし てのボリュームラベルを配送する32バイトの入 口を収容している。

ルートディレクトリの後の残余のポリュームは、 クラスタのブールとして見ることができるファイ ル部域(各ファイル領域には1つ以上の論理的セ クタが収容されている)として知られている。各 てテーブルの後にはルートディレクトリ(root directory) が続き、このルートディレクトリの後にはポリュームファイルが続く。ブートセクタには、ブートバラメータブロックすなわち898 と呼ばれる。戦る領域におけるポリュームに関する様々の記述情報(descriptive information)、ドライブの数及びポリューム1.0、のような情報、及びブートストラップルーチン(bootstrap routine) が収容されている。

ファイル割当でデーブルは、ディスクについての割当て可能なクラスタ(これらのクラスタは、セクタを2楽したものである)に直接相当するフィールド(欄)に区分される。一般に、これらのフィールドは16ビットの幅を有している。最初にリザーブされたFAT 人口には、BFB においても異出すことができるメディア記述バイト(media descriptor hyte)のコピーが収容されている。リザーブされた残余のフィールドにはBFFBが収容されている。残余のFAT 人口には、それらの相当ディスククラスタ(corresponding disk clasters)

クラスタは、PAT の対応人口(該人口には、FAT の現在使用。すなわち、利用できること、リザー アされていること、ファイルに割り当てられていること、又は使用できないことが記述されている) を備えている。

FAT ファイルシステムは、1m以下のボリュームで優れた性能を得ることができる。しかしながら、ボリュームのサイズが1mを超えると、FATファイルシステムの性能は急激に低下する。容易に入手可能なハードディスクのサイズは急激に増大しているため、このことは重要な問題となっている。

ボリュームが18b以下の場合には、FAT は、いつでもランダムアクセスメモリ内に保持される程
充分に小さく、従って、ファイルのいかなる部分
にも非常に高速のランダムアクセスを行うことが
できる。しかしながら、ハードディスク又は固定
ディスクに適用した場合には、FAT は大き過ぎて
メモリに保持できなくなり、かつ紹分してメモリ
内にベージ付けしなければならない。このため、

多くの余分なディスクペッド運動が必要になり、 コンピュータンステムのスループットを低下させ ている。また、ディスクの空きスペース(free a. pace) についての情報が、PAT の多数のセクタを 機切って分散されるため、連続的にファイルスペ ースを割り当てることは実際的でない。このため; ファイルが銀分化され、コンピュータシステムの スループットが更に低下される。また、ハードディスクに比較的大型のクラスクを用いるため、無 駄なスペースが非常に大きくなる。

第5A図~第5H図には、設置可能なファイル システムの1つの場合のディスクフォーマットを 示す一連のダイアグラムが示されている。このファイルシステムは、高性能ファイルシステムは、高性能ファイルシステム(Single performance file system、HPFS) と呼ばれているものである。本発明の高性能ファイルシステムは、FATファイルシステムについての上記問題を解消でき、かつあらゆる形式のディスクメディアについて優れた性能を発揮できるものである。 第5A図に示すように、HPFSボリュームは、前に

ブートブロック (BootBlock) 5 8 2 の後には~ スーパープロック (SuperBlock) 512及びスペ アブロック (SpareBlock) 514が続いている。 スパープロックSi2は、ディスクメインテナン スユーティリティによって変更されるに選ぎない。 スーパープロック312は、空きスペースのビッ トマップを指すポイング316、バッドブロック リスト518、ディレクトリブロックバンドを指 すポインタ520、ルートディレクトリを指すポ インタ522を収容している。更にスーパープロ ック512は、デート (日付け) を構えたデート フィールド (日付け機) 524を収容しており、 ポリュームは、CHKBSKにより最終チェック及び棒 復かなされる。C88958は、ディスクの懸い部分を 検出しかつカタログするための良く知られた08/2 ディスクユーティリティである。

スペアブロック 5 1 4 は種々のフラグ及びポインタを収容しており、これらについては以下に詳述する。スペアブロック 5 1 4 は、コンピュータンステムが実行されるときに変更される。

形成されたPAT バーティション形の側面に沿って 固定ディスク上に設けることができる。BPPSボリュームは、512バイトのセクタサイズを使用し ており、2199Gb (2**のセクタ) の最大サイ ズを有している。BPPSは固定ディスクに使用する ことを主として設計されているが、実際上、あら ゆる形式のディスクメディアとの互換性を有して いる。

BPPSボリュームは、固定された構成が発ど必要とされない。アートブロック502にはボリューム (8 8b) のセクク0~15が割り当てられ、該セクタ0~15は、ボリュームのネームフィールド (名前職) 504、32ビットボリュームの18フィールド 506、8105パラメータブロック508、ディスクブートストラッププログラム 510を収容している。ディスクブートストラッププログラム 510は、オペレーティングシステムファイルが設出される限りは、これらのオペレーティングシステムファイルの位置付け及び鉄取りを行う限定モードで使用することができる。

残余のボリュームは、ファイルの記憶に使用さ れる886パンド (例えば、バンド516~522) に区分される。第5人間には4つの886パンドが 示されているが、HPFSは非常に多数のバンドを得 ることができる。各バンドには、それ自体の空き スペースピットマップ (横先ば、ピットマップ 524~534齢服)が設けられている。空きス ベースピットマップの各ピットは、セクタを衰し ている。セクタが使用されている場合にはピット は0であり、セクタが使用可能(アベイラブル) であるときは、ビットは1である。ビットマップ は、バンドのヘッド又はテールに位置付けされる ため、2つのビットマップは交互のバンドの際に 隣接している。ビットマップのバンドサイズは、 任意のサイズのファイルを収容できるように変更 できるけれども、上記構成により、16mbになる ようにファイルに割り当てることができる最大の 連続望きスペースを得ることができる。ディスク のシークセンタにおいて(又はシークセンタに例 かって) 拉翼付けされた1つのバンドは、ディレ

クトリブロックバンドと呼ばれ、後述するような 特別の処理を受ける。

#FFSの全てのファイル又はディレクトリは、第 38図及び第5C図に示すFnode と呼ばれている 基本ファイルシステムの目的(オブジェクト)に アンカーされている。Paode 538は、ファイル 又はディレクトリに割り当てられた最初のセクタ すなわち第1セクタであれ、スーパーブロック 504におけるフィールド522により指示され ている。各Frode 530は単一のセクタを占拠し、 かつ、第58回に示すように、ファイルシステム により内的に用いられる網御及びアクセス情報フ ィールド540、拡大器性 (extended attribute、 「EA」) 及びアクセス制御リスト(access control lists、「ACLa」)を配位する額級542、關係 するファイル文はディレクトリのネームの畏さ及 び最初の15文字を表示したい場合にはそのため のフィールド514、及び割当て構成546を収 容している、frode は、これを代表するファイル 又はディレクトリの近くに常に記憶されている。

探用しており、8回以上の実行を有している。Pnodeの割当ては、割当てセクタのB+つりー(本)
のルート(程)となり、このルートには、第5日
個に示すように、ファイルのセクタ実行を指す実際のポインタが収容されている。B+つり一及び
B-つりーの概念については後で詳述する。Pnodeのルートは、12のエレメントのためのルーム
を有している。各割当てセクタは、種々のセクタ
情報に加えて、セクタ実行を指す40個程のポインタを収容することができる。従って、本発明の
好ましい実施例においては、2レベル割当ての
B+つりー(iwo level allocation B+ Tree)が、
7.68 6h (12*40*) 回のセクタ実行
のファイルを記述することができる。

これとは異なり、高度に細分化されたファイルを記述するには2レベル割当でのB+ツリーが充分でない場合には、BPFSファイルシステムが、ツリーに必要なだけの付加的レベルを導入する。中間レベルにおける割当でセクタは、60個程の内

第5 C図に示す割当て構成 5 4 6 は、ファイル 又はディレクトリの連続性のサイズ及び度合に基 づいて機つかの形態をとる。本発明のBPPSは、1 つ以上の連続セクタの1つ以上の実行(rons)又 はエクステントの割当てとして、ファイルをビュ ー(v)ews)している。各実行は、1 対の二難ワード、すなわち、セクタにおける32 ビットのスタートセクタ数及び32 ビットの長さ(この長さは、 実行長さエンコーディングと呼ばれている)により配号化される。アプリケーションプログラムの 観点からすると、エクステントは目で見ることは できない。すなわち、ファイルはバイトの継ぎ目 のない流れであると考えることができる。

Freds における割当て情報にリザーブされたスペースは、各16%までのセクタの8回の実行と 問数のポインタを保持することができる。従って、高度連続サイズのかなり小さなファイルを、Fredeの中に完全に記述することができる。

#FFSは、Fnode にとっては大き選ぎるか細分され過ぎているファイルの位置を示す新しい方法を

部(端部ではない)B + ツリーノードを保持する ことができ、このことは、この構成の記述能力が 極めて大きな数に急速に成長することを意味して いる。例えば、3 レベル割当てB + ツリーは、 28,800 (i 2 * 6 0 * 4 0) 囲のセクタ実行を記述することができる。

割当てセクタの実行長さエンコーディング(renterestate encoding)及びB+ツリーは、ファイルのサイズ及び位置を充分に特定できるメモリであり、かつ従来技術に比べ幾つかの優れた長所を有している。セクタ数への論理的ファイルオフセットの翻訳は極めて高速に行われる。すなわち、ファイルシステムは、正しい範囲が見出されるまで実行サイズを要約し、実行ポインタ(run pointers)のリスト(すなわちリストのB+ツリー)を単に模切るだけである。そのとき、簡単な針算を行うことにより、実行の中でセクタを識別することかできる。また、新たに割り当てられたセクタがファイルの前の最終セクタと連続している場合には、実行長さエンコーディングによって、ファ

イルを論理的に拡大することが極めて簡単になる。 このファイルシステムは、ファイルの最終実行ポ インタのサイズ二重ワード (size dosble-word) を単に増大させるだけであり、適当な空きスペー スのビットマップにおけるセクタのビットをクリ アするように構成されている。

ファイルと同様に、ディレクトリはFoodesにアンカーされる。ルートディレクトリ(root directory) についてのFoode を指すポインタは、スーパープロック512において見出すことができる。 数5日間は、本発明によるディレクトリ構成を示すものであり、ここにはディレクトリFoode 550 が示されている。ルート以外のディレクトリについてのFoode は、それらの数ディレクトリ(parent directories) におけるサブディレクトリ人口を通って到達する。

ディレクトリは、ディスク上にもつの選択セクタ (consecutive sectors)として割り当てられる 2 % hのディレクトリブロックから構成されていて、 任意のサイズに成長することができる。例えば、

578と、日-ツリーボインタを収容するフィールド58日とが含まれている。各ディレクトリ入口は、入口の長さを含んでいるワード582で始まる。これにより、各入口の終時におけるフレックススペースの可変量が与えられる。この可変量は、ファイルシステムの特別なバージョンに使用できるようにし、かつディレクトリブロックを極めて迅速に模切らせることを可能にする。

ディレクトリブロックの入口の数は、ネームの 長さによって変化する。平均的なファイルネーム の長さか 1 3 文字であるときには、平均ディレク トリブロックはほぼ 4 0 個の入口を保持するであ ろう。ディレクトリブロックの入口は、それらの ネームフィールド(名前欄)の2 進字句順序(binary [exical order] により分類される。最終入 口は、ブロックの終了を印すダミーレコードであ る。

ディレクトリが大きくなり過ぎて1つのブロック内に記憶されなくなった場合には、B-フリー として編成される2×5ブロックを付加することに ディレクトリブロック 5 5 2、 5 5 4、 5 5 6 を 参照されたい。このファイルシステムは、ディス クのシークセンタ又はその近くに位置付けされた ディレクトリバンドにディレクトリブロックを翻 り当てることを試みている。ディレクトリバンド が満たされると、スペースが利用できる限り、デ ィレクトリブロックが割り当てられる。

2xbの各ディレクトリプロックには、1つから 多数のディレクトリ人口を設けることができる。 例えば、入口558~568を参照されたい。ディレクトリ人口には幾つかのフィールド(ない 第5日間に 深すように、時間及び日付スタンプのためのフィールド570と、Frode ポインタを収容するフィールド570と、Frode ポインタを収容するフィールド570と、ディスクメインテナンスプログラム(このプログラムは良く知られたものである)による使用ができるようにするための用法カウントフィールド(usage count field)574と、ファイル長さすなわちディレクトリネームを収容するフィールド576と、ネーム目体のフィールド

よりサイズを大きくできる。特定のネームをサーチする場合には、ファイルシステムが一致を開出すか、目的とするネームよりも字句的に多いネームを見出すまで、ファイルシステムがディレクトリブロックを横行する。後者の場合、ファイルシステムが、入口からBーフリーのポインタを抽出する。このポインタがどのサーチフィールドをも指示しない場合には、ファイルシステムは、フリーにおける次のディレクトリブロックを次のポインタにより指示させ、サーチを統行する。

プロック当り40個の入口があるものと仮定すれば、ディレクトリプロックの2レベルツリーは1,640個のディレクトリ入口を保持でき、3レベルツリーは55,640個の入口を保持することができる、機器すれば、最大限3つのディスクアクセスを構えた一般的な65,640個のファイルにおいて、特定のファイルを見出すことができる(又は、それが存在しないことを示すことができる)。ディスクアクセスの実際の数は、キャッシュコンテンツ(cache contents)及びディレクトリプロック

のB-ツリーのファイルネームの位置に基づいて 定められる、これは、最悪の場合4.000 個のセク タを読み取って、同数のファイルを収容している ディレクトリにファイルが存在するか否かを簡定 しなければならないFAT ファイルシステムに対し て顕著な改善を与えるものである。

#PPSのB-つリーディレクトリ構成は、開放作業及び見出し作業に関するその効果を超える興味ある含意(isplications)を有している。ディレクトリブロックを付加(又は除去)するか、成いはネームが或るブロックから他のブロックに移動されてツリーのバランスを保つとき、ファイルの創出、リネーミング又は制除により。複雑な作業のカスケードを生じさせるかもしれない。実際、ファイル自体が成長することはないけれども、リネーム作業によってディスクスペースの不足をきたすであろう。この問題を回避するには、#PPSが、ディレクトリの緊急時に引き出すことができる空きブロックの小さなブールをリザーブし、このブールを指すポインクがスペアブロックに記憶され

B+ツリーが創出されて実行が記述される。単一のBAが非常に大きい場合には、数SAはPacdeの外に押し出され、該BA自体のB+ツリー内に押し込まれる。

本発明は、アブリケーションプログラムがファ イルの拡大機性 (extended attributes)を走蔵す ることを可能にするOS/2カーネルのAP:機能、す なわち、9089Filalafa及びBasSetFileInfoを改善 することができる。また、本発明によれば、任意 のパスネーム(パス名)と関連する84s の顕取り 及び書込みに使用できる2つの新たな機能、すな わち、BOSSPathInfo及びBosSetPathInfoを得るこ とができる。アプリケーションプログラムは、特 定のEs(これは、一致させるべきネームを供給す る)のバリューを要求するか、ファイル又はディ レクトリについての全てのBis を一度に得ること ができる。BAS の支持により、目的にかなったア ブリケーションブログラムの使用が容易になる。 ファイルを所有するアプリケーションのネームか ら、從蹊ファイルのネーム、アイコン及び実行コ

るように構成するのが好ましい。

ファイル選性は、ファイルの明白な記憶領域の 外でオペレーティングシステムにより維持される ファイルについての情報である。

本発明のHPFSは、拡大機性(Extended Attributes 、「EAs」) を支持し、

ネームギバリュー

の形態をとる。但し、バリュー部分は、ゼロで終 わる配号列(null-terminated string、「ASCITZ」) でもよいし2進データでもよい点を除く。本発明 の好ましい実施例においては、各ファイル又はディレクトリは、これに取り付けられたFAs の 6 4 Kbの最大値をとることができる。個し、この制限 は容易に変更することができる。

EAS の記憶方法は変えることができる。所与のファイル又はディレクトリと関連するEAS が充分に小さい場合には、これらのEAS はFnode に記憶されるであろう。また、EAS の全体のサイズが非常に大きいときには、これらのEAS はセクタ実行においてPnode の外に記憶され、割当てセクタの

ード(executable code) に至る然と全ての形式の 情報をEds に記憶させることができる。

WOSは、多レベルでのディスクスループットの 潜在的ボトルネックをアタックする。性能を向上 させるため、IPFSは、進歩したデータ構成、連絡 セクタ劉当で、インテリジェントキャッシング、 統取りヘッド、及びデファード書込み (deferred writes)を使用している。最初に、MPFSは、その データ構成、すなわち、ファイルキーム、ディシ タドリネーム、ファイル又はディレクトリに割り 当てられたセクタのリストへの高速ランダムアク セスが行えるようにした複雑なデータ構成(Bー ツリー及び8+ツリー)、及び適当なサイズの空 きスペースのチャンクを位置付けできるようにす る簡単でコンパクトなデータ構成 (ビットマップ) をタスクに一致させる。これらのデータ構成を探 作するルーチンは、アセンブラ営語で記載するの が経ましい。

#PFSの主目的は、可能な限り、連続セクタをファイルに割り当てることである。ディスクの談取

り/書込みへッドを収るトラックから他のトラックに移動させるのに要する時間は、可能性のある他の遅延よりも遙かに重大であり、このため、IPPSは、ファイルスペースを連続的に割り当てることにより、及びFrode 及び該Frode が制御する事柄の近くの空きスペースピットマップのような制御機を維持することにより、このようなへッド運動を囲避するか最小限にする。高度の連続的なファイルはまた、多くのセクタに対し一度に要求されるディスクドライバのリクエストを、ファイルシステムによって少なくすることを補助し、ディスクドライバが、ディスクコントローラの多セクタ移送能力を活用できるようにし、かつ、修理すべきディスクの完全な中断数を低減させることができる。

多数のファイルを同時に更新させるマルチタス キングオペレーティングシステムにおいてファイ ルが細分化されないように維持することは、従来 技術には見られない特徴である。#PPSが用いてい る1つの方法(手順)は、新しく創出されたファ

頼って、HPFSが要求する物理ディスク転送の数を 最小限にしている。HPFSは、FAI ファイルシステ ムが行ったようにして、セクタのキャッシングを 行う。しかしながら、FAI ファイルシステムとは 異なり、HPFSは非常に大きなキャッシェを効率 く管理し、セクタキャッシングを、パーハンドル ペース(per-handle basis)で、ファイルが用いら れる方法に調節するようになっている。また、 HPFSは、パスネーム、ディレクトリ、トランスフ ォーミングディスクディレクトリ入口を、配憶表 現(memory representation) におけるよりコンパ クトで効率の扱いものにキャッシングする。

性能を向上させるべくBPFSが用いられているもう1つの技術は、プログラムが必要とすると考えられるデータを予め扱み取ることである。例えば、ファイルが開かれるとき、ファイルシステムが、Fnode 及びファイルの内容の最初の幾つかのセクタを予めよみとりかつキャッシングする。ファイルが、該ファイル住の実行プログラム(executable program) 又はヒストリー情報である場合には、

イルを削々のバンドのディスクを模切って分散させ、できるならば、セクタが拡大されるときファイルに割り当てられるセクタがインターリーブされないようにすることである。他の方法は、ファイルを拡大しなければならない復報に、連続スペースの444をファイルに予め割り当てて、ファイルを閉じるときに全ての過剰のスペースを戻す方法である。

アブリケーションが、新しいファイルの最終的サイズを予め知ることができるならば、BPPSがファイルを創出するときに、最初のファイル割当てを特定化することにより、BPPSを補助することができるであろう。そうすれば、システムは全ての空きスペースピットマップをサーチして、ファイルを充分に保持できる連続セクタの実行を見出すことができるであろう。このことに失敗した場合には、システムは、ファイルのサイズの1/2 である2ラウンドをサーチし、以下このことが繰り返される。

8785は、幾つかの異なる種類のキャッシングに

Prode は、全ファイルを確ちに連続的に読み取ることにより、ファイルの開放作業が一般的に続けられていることを示す。ファイルシステムは、より多くのファイル内容物を用意しかつキャッシングする。プログラムが比較的少量の級取りりクエストを発行する場合には、ファイルシステムは28bのチャンクのファイルから絶えずデータを取り出し、過剰のデータをキャッシングする。このキャッシングにより、殆どの読取り作業が満足できるものになる。

本発明のBPFSは、OS/2のマルチタスキング能力 に基づいた遅い審込み(lazy writes、デファード 審込み又はライトビハインド(write behind) と も呼ばれている)を綴りにしているところが大き い。例えば、プログラムがディスク書込みを要求 する場合には、データはキャッシュ内に置かれ、 キャッシュバッファかダーティとしてフラグされ る (すなわち、ディスクのデータの状態と一致し ないことを示す)。ディスクがアイドル状態にな るか、祓いはキャッシュがダーティバッファで飽 和されると、ファイルシステムは、グエモンプロセス (daeson process) からのキャプティブスレッド(captive thread)を用いてバッファをディスクに覆き込み、最も古いデータでスタートする。キャプティブスレッド及びダエモンプロセスについては、Hastings、その他の著者によるテキストシリーズ「マイクロソフト社の05/2プログラマーズリファレンス(*Hicrosoft O5/2 Programmers Reference*)」(1989 年、Hicrosoft Press 社刊)において説明されている。

一般に、遅い苦込みは、プログラムがより高速 で実行されることを意味している。なぜならば、 一般に、プログラムの読取りリクエストは、書込 みリクエストを待機して遅延することなく完了す るからである。綴り返し読み取られるプログラム の場合は、小さなワーキングセットを表更して鬱 き込むので、遅い審込みはまた、多くの不必要な すなわち冗長な物理ディスク鬱込みを問避するこ とができる。遅い鬱込みはそれらの疲る危険性を 有しており、従って本発明は、BosOpen に対して

verification)中、ハードウェアのディスクドラ イバにより検出される。

書込みエラーを取り扱う主要機構は、ホットフィックス(hotfix)と呼ばれる。エラーが検出されると、ファイルシステムは、リザーブされたホットフィックスプールから空きブロックを取り出し、該ブロックにデータを書込んで、ホットフィックスマップを更新する(ホットフィックスマップを更新する(ホットフィックスマップを更新する(ホットフィックスマップのであり、二重ワードの各対は、そのホットフィックス交換の番号と関連する番号の悪いセクタを収容している)。次に、ホットフィックスマップのコピーがスペアプロックに書き込まれ、ディスク装置に問題があることをユーザに知らせる警告メッセージがディスプレイされる。

ファイルシステムがディスクドライバからのセクタ誘致り又は書込みを要求する度毎に、ファンシステムは、ホットフィックスマップを走査して、 悪いセクタの番号を実際のデータを保持している ほいセクタに相当する番号に置き換える。 GpenModeパラメータのライトスルーフラグ(write -through flag)を設定することにより、パーハンドルベース(per-handle basis)上で遅い書込みに打ち勝ち、BosBufReset 機能により、パーハンドルベース上のディスクにデータを委託(coseit)することができる。OS/2の現行パージョンにおいても、BosOpen 及びBosBufReset の顕微能を利用することができる。

選い費込み(iazy write)の広範囲な使用により、NPPSが、あらゆる繁急事態の下での責込みエラーから優雅に回復できるようになる。例えば、
数込みの失敗が知られるときまでに、アプリケーションは長時間を要する。なぜならば、アプリケーションは、データをディスク記憶装置内に安全に選び出したという錯貨の下で行われるからである。ディスクアグブグにより戻される「セクタが見つけないエラー("sector not found" error)」のようなエラーは、ハードウェアにより検出することができる。或いは、そのようなエラーは、データの審込み後、疏取り検疑(read-after-write

CBKDSNの1つのデューティはホットフィックスマップを空にすることである。ホットフィックスマップの各交換ブロックに対し、CBKBSKは、データを所有するファイルに対する好ましい位置にある新しいセクタを割り当て、データをホットフィックが新しく割り当てられたセクタに移動させ、かつ、ファイルの割当て情報(この情報には、再バランスしている割当てツリー及び他の精巧な作業が含まれている)を更新する。改いで、CBKDSKは、窓いブロックリストに悪いセククを行加し、交換セクタを解放してホットフィックスプールに戻し、ホットフィックスマップをスペアブロックに書き込む。

#FFSは、各#FFSポリェームのスペアプロックに グーティFSフラグを維持する。#FFSポリュームの 全てのファイルが閉じられるとき、キャッシュ内 の全てのグーティバッファが書き込まれるとき、 或いは、ブートポリュームの場合にはシャットグ カンが選択されかつその作業を完了したときに、 フラグがクリアされる。

05/2のブートシーケンス中に、ファイルシステムが各8PPS上のダーティPSフラグを検査して、フラグが設定されている場合には、CBKBSKが実行されるまでブートボリュームには更にアクセスできないようにする。ブートボリュームにダーティPSフラグが設定されている場合には、システムは自動的にCBKDSKを実行する。

スーパープロック又はルートディレクトリの損失というような真に重大な大事故の場合には、最も成功の可能性のあるデーク回復を与えることができるようにHFPSが設計されている。Fnode。割当でセクタ及びディレクトリブロックを確えたかと全ての形式の重要なファイル目的(オブジェクト)が、その観及び子の両方に二重にリンクされており、かつ、ユニークな32ビットのサインを収容している。Fnode はまた、それらのファイル又はディレクトリのネームの最初の部分を収容している。従って、SHOBS は、Fnode、割当でセク

データの特定ビースを見出すには、2選ツリーがルートを模切るようにする。各ノードにおいて、所望のキーがノードのキーと比較される。両キーが一致しない場合には、所望のキーがノードのキーより小さいか大きいかに基づいて、ノードのサブツリーの1つのブランチ又は他のブランチが選択される。このプロセスは、一致が見出されるまで、又は第5下図に示すように空のサブツリーに出合うまで続けられる。

このような簡単な 2 進ツリーは、選解と実施が 容易であるけれども、実用に際しては欠点を有し ている。キーが非ランダムな態様でツリーに首尾 良く分散されなかったり付加されなかったりする と、ツリーが全く非対称的になり、ツリーの機類 時間が広範囲に変化してしまう。

アクセス時間を均一にするため、多くのプログ ラマは、第5回に示すようなB-ツリーとして知 られているバランス形ツリーを好む傾向にある。 B-ツリーについての重要な点は、データが全て のノードに記憶され、1つ以上のデータ項目が1 ク及びディレクトリブロックに対してディスクを 規則正しく走査し、Foode 、割当てセクタ及びディレクトリブロックを用いてファイル及びディレクトリを再構成し、かつ最後に空きスペースのビットマップを再創出(regenerating)することにより、全ポリュームをリビルドすることができる。

上記のように、本発明は、ファイル及びディレクトリを論理的に順序付けするのに、B+ツリー及びB-ツリー (2 進ツリー) を用いている。 2 進ツリーは、データを物理的に順序付けすることなくして、ポインタを用いてデータ項目の集合を論理的に順序付けする技術である。

係 5 F 個を参照すれば、簡単な 2 建ツリーにおけるタードには、ツリーにおけるノードの論理 的位置を決定するキー値を含む残るデータと、並 びにノードの左右のサブツリーを指すポインタと が設けられている、ツリーを開始するノードはル ート(根)として知られており、ツリーの技の端 部に位置するノードは、ときどきリーフ (葉)と 呼ばれている。

つのノードに記憶され、かつ、ツリーの全てのブ ランチが綴じ長さをもっているということである。

B-ツリーの最悪の場合の挙動(behavior)は 予捌可能であり、簡単な2次ツリーの挙動より遙 かに良好であるが、B-ツリーのメインテナンス はかなり複雑である。新しいデータ項目の付加、 キーバリェーの変更、又はデータ項目の削除によ り、ノードのスプリッティング(分割)又はマー ジング(併合)が生じ、これにより、ツリーには 他の作業のカスケードが強制される。

第5G図に示すように、B+ツリーは、2つの ノード形式 (内部ノードは他のノードを指すだけ であり、外部ノードは実際のデータを有している) をもつB-ツリーの特殊化されたフォームである。

B-ツリーよりもB+ツリーの優れている点は、B+ツリーの内部ノードが、B-ツリーの中間レベルノードより非常に多くの決定バリューを保持でき、そのため、ツリーの外のファンが高速になりかつブランチの平均長さが短くなることである。これにより、必要データを見出すにはB+ツリー

のブランチがその端部に統かなければならないという事実を補償でき、一方、B - ツリーにおいては、データは中間ノードにおいて発見され、厳いは、ルート(相)においてさえも発見される。

本発明は、03/2オペレーティングシステムを改 寄したものであり、多くのユーティリティとOS/2 の現行バージョンにおいて利用できるサブルーチ ンを用いて実施することができる。本発明は、主 としてOS/2オペレーティングシステムに使用する ことを意図しているが、本発明の原理は、実際の あらゆるコンピュータのオペレーティングシステ ムに適用できるものである。ここで説明する新し いユーティリティ及びサブルーチンを除き、他の 全てのユーティリティ及びサブルーチンは現在利 用されていて食く知られたものである。98/2オペ レーティングシステムの詳細な説明については、 前途の0S/2ブログラマ用最考賞を参照されない。 本発明の改善された05/2オペレーティングシステ ムのボリュームマネージメント (volume manage-*ent) は、OS/2の従来のバージョンにおいて行わ

ュームの管理のために過去に設けられた全ての FSDsに対して同じサポートを提供する。本発明は、 強く知られている既存の05/2呼び出し(05/2 cal-1s) 並びに以下に説明する幾つかの新しい機能を 有している。本発明の設置可能(insis | lable) な ファイルシステムについての完全な説明は、マイ クロフィッシュの形態で本額に添付されかつ参考 として掲示する付録 1 (appendix 1) において述 べられている。

本発明は、傷々のボリュームについての正しい ファイルシステムドライバの識別及びローディン グが容易に行えるマウントプロセス及びアンマウ ントアロセスを用いることを衰弱している。

マカントプロセスは、幾つかの異なる事象が生 じたとき、すなわち、

ボリュームへの最初のアクセスがあったと
 ま。

2. ドライブのボリュームが不確実になる (このことは、通常、ユーザが新しいメディアをドライブに入れることを意味する)全てのとき。

れているものと同じデューディ、すなわち、悪いボリュームがドライブにインサートされたときの検出、ボリュームが除去されたときの検出、ボリュームバラメークブロック(VPB)を介してドライブ内に置かれた新しいメディアに関する新しい情報の創出、適当なデバイスドライバとの適信、新しいインサートメディアにアクセスする必要のあるデバイス情報をシステムに与えること、バッファ及びCOS機構とのインターフェース、及び特定ボリュームへの変更をシステムに知らせること等に応答することができる。

OS/2の従来のバージョンにおいては、僅かにし つのファイルシステムがあったに過ぎない。本発 明によれば、統一された環境内に多数のファイル システムを設けることができる。ボリュームマネ ージャは、どのファイルシステムを特定のボリュ ームにアクセスさせるべきかを決定し、ファイル システムドライバ (FSDs) が特定のボリュームに ついてのそれらの質測(resources) を管理 (マネ ージ) できるようにする機構を提供し、かつボリ

3、ドライブ内にないポリュームへのアクセス が要求される全てのとき、 に開始される。

マウントプロセスへの入力は、ドライブバラメータブロック (OPB) (このドライブバラメータブロックは、デバイスドライバにI/O を行うこと、及びドライブ内にあると現在考えられているボリュームのVPB にハンドルを記憶させることに使用される) を指すポインタである。マウント作業によりこれが更新される。ローカルVPB がスタック上に割り当てられ、OPB ポインタと共に初期化される。

第6 図に示すように、項目602で示すように メディアの論理的セクタ0を読み取ることにより、 マウントプロセス600が開始される。デバイス ドライバからの全てのエラーは無視する。なぜな らば、異なる形式のメディア(すなわち、光学ディスク又はCB-ROM)がトラック0を読取り不能に できるからである。論理的セクタ0を読み取る前 に、テンポラリマウントバッファが0に初期化さ れる。ボリュームのラベルテキストフィールドが
「UNLABBLEB」に初期化される。セクタロがチェ
ックされ、特定バリュー(41)に対してサイン
バイト (signature byte) を比較することにより、
フォーマットが認識されているか否かを決定する。
フォーマットが認識されていない場合には、VPB
に近い情報がスタック上に充壌される(すなわち、
3 ミビットボリューム連続番号(32 Bit Volume
Serial Bumber))。

次に、項目604により、BBILDBPB呼び出し
(80]LDBPB call) が、BPB において特定化された
デバイスドライバに発行される。BBILDBPBは、デ
バイスドライバによりエクスポート (移出) され
る手駆である。このBDILDBPB手類については、
付録1において詳細に説明されている。BUILDBPB
は、装置の物理バラメータ (バイトバーセクタ
(byte per sector)、セクタバートラック(sector per track) 等) を学習させるべく呼び出され
る。デバイスドライバは、デバイスドライバは、
ポリュームの物理バラメータを決定するのに用い

すポインタを購えたFS_MOUNT(フラグ× O)の入 口点を呼び出すことにより、各PSD をポール(po-11) する。850 はPSIL DoVol10 を呼び出し、#リ ェームから他のセクタを読み取る (それ自体のバ ッファを割り当てなくてはならない)。 PSB が、 ERBOR_UNCERTAIN_MEDIA に戻る場合には、エラー ・が戻され、プロセスは、決定(dacision) 6 1 8 G より示すように再スタートされる。 888 がブート セクタをサポートする場合には、FSD は、ブート セクタのファイルシステムのネームフィールドを チェックして、これがネームフィールドを認識し ているか否かを決定する。PSB がブートセクタを サポートしない場合には、PSD がポリュームを認 識しているか否かを決定すべく。装置へのI/O が 行われる、PSD がひとたびポリュームを認識して いるならば、項目612で示すように、898ファ イルシステムの独立及び従際領域における適合フ ォールドを更新する。VPS ファイルシステムの独 立及び従属領域については、第7回に関連して栗 に詳細に説明する。この時点においては、FSD は

ることができる情報を収容しているパップナを指すポインタに導かれる。殆どのドライバにとっては、これはセクタのであり、非常に古い幾つかのドライバにとっては、PATの最初のセクタである。装置が、セクタのから読み取られるデータを解釈できない場合(例えば、この場合のフロッピがPATではなく、従ってPATIBバイトが意味をもたない場合)には、装置は最小のBPSを探し、カーネル及びFSOsが必要なI/Oを行って、ポリェームを完全に機関できるようにする。

前に創出されたBPB からの適合フィールド(relevant lield) は、スタックのローカルVPB (すなわち、Sectors/track、MumberolBeads、Total Sectors、Sector Size)にコピーされる。新しいVPB か割り当てられ、ローカルVPB からの情報がそれにコピーされる。次に、本発明によれば、ループ60m入り、項目60mで示すように、新しく創出されたVPB、論理的セクタのを指すポインタ。及びVPB ファイルシステムの独立及び従

FS Reiper(FSB)機能を発行して、新しいボリュームが、本発明が管理する他の任意のボリュームと同じであるか否かを決定する。このFS Reiper は、ファイルシステムの独立及び従属領域にボインタを関す。次いでFSB は、項目 6 1 4 で示すように、新しく創出されたVFB から古いVFB へと情報をコピーする。新しく創出されたVFB は、ROUNT の呼び出しを行った後に破壊される。次に、FSB は、あらゆるバッファを無効にするような古いVFB に対してあらゆるクリナップ作業(civanup work)を行う。これは、ドライブからボリュームが除去されていることがあるからである。

本発明によれば、ひとたびFSB がポリュームを 認識すると、リスト内に一致するものが見出され る場合には新しいVPB が除去される。リスト内に 一致するものが見出されない場合には、VPB は。 マウントされたFSBsのリストにリンクされる。 FSDsが認識されない場合には、決定614及び項 目616に示すように、VPB が空にされかつPAT ファイルシステムがマウントされる。 新しいポリュームがドライブにインサートされかつ古いポリュームに対してカーネルがもはや関与しない場合には、本発明では、FSD にFS_MOUNT (フラグニ2) が発行され、これにより、このポリュームに割り当てられた資源の割当てを解除するようになっている。

本発明は、FSD により要求される機能に対する 既存のカーネル質額を利用するのに、効率の良い

のヘルバを使用して、要求されたポリュームが実 際にドライブ内にあることを保証し、適当なデバ イスドライバを呼び出し、かつハードエラーを取 り扱う。このヘルバは、PSB 内で常時用いること ができる、FS_MOUNT 呼出しの範囲内で呼び出さ れるとき、FSD はドライブのボリュームに適用さ れる。しかしながら、FSO がFS MOORT 呼出しに **戻るまでは、ポリュームの認識が完了していない** ので、ERRON UNCERTAIN MEDIA が戻される場合に は、FSD に注意しなければならない。このことは、 ドライブにおけるメディアの畿別を試みる間に、 メディアが不確実になっていることを示すもので ある。また、このことにより、PSB が認識を試み ていたボリュームが除去されたことを示すように することもできる。この場合には、PSB は、FS___ BOOM 呼出しに導かれたWPBに取り付けられた (attached)全ての資源を解放し、ERNOR UNCERTA-IN MEDIA がFS MOUNT 呼出しに異される。これ により、ボリュームトラッキング論理が、マウジ

トプロセスを再スタートするように命令される。

機構を用いることを意図するものである。より詳しくは、FSBがカーネル内に存在する機能を要求する場合には、FSBは、ファイルシステムへルバ(FSH)を呼び出す(invoke)ファイルシステムへルバ呼出し(call)を発行する。呼び出されたFSBは、次に、要求された情報を関す。以下に、ファイルシステムへルバについて簡単に説明する。以下に述べる要約においては幾つかの重要なファイルシステムへルバがリストアップされているけれども、必要に応じて付加的なファイルシステムへルバを設けることができる。ファイルシステムへルバは、&ppendix 1 において詳細に説明されている。

ファイルシステムヘルパ (File System Helpers)

FSH_GETVOLPANN: 多くのFS呼出し時に、VPB へのハンドルがPSB に導かれ。FSB が、VPB のファイルシステムの独立及び従落領域にアクセスすることかしばしば必要になる。このヘルバは、そのようなサービスを与えるものである。

FS<u>H_00</u>90110 : PSD が、特定のポリュームに対 して1/0 を行う必要があるときには、FSD は、こ

FSII_BUPLICATE VPB: FS__HOURT 呼出しの間、入力VPB は、管理されている他のしつのボリュームと同じボリュームにすることができる。新しいボリュームに関する更新情報を創出しかつ古い複製VPB(elder duplicate VPB)への情報をコピーすることは、FSB の責任である。このヘルパは、古い複製VPB が存在しているか苦かを決定し、存在している場合には、古い複製VPB のファイルシステムの独立及び提展領域を指すポインタを関し、これらの領域がFSD によって更新されるようにする。次いで、FSD は、ボリュームが除去されているので、古いボリュームについてあらゆるクリナップ作業を行う。

上記のように、本発明は、可能な限り、予め存在している05/2資源を使用することを狙ったものである。以下のリストは、本発明の作業中に呼び出される機能の階層(bierarchy) を要約したものである。

1 Oovallo

1.1 WhatVolume

特開平3-171238 (18)

LLI	ProbeChange	1.1.3.13	Unlock¥Buf
1,1,2	VasatMedia	1, 1, 3, 14	Bullavalidate (Redeterm-
1.1.3	Genhyph		(sibah sai
1.1,3.1	LockYBuf	1.1.3.15	FlushBuf (Redetermine M-
1.1.3.2	ReadBook		edia)
1.1.3.3	84841148	1.1.4	Incapage
1.1.3.4	FSKountValume	1,1,5	BecVPBRef
1.1.3.4.1	3*g_Cet	1.1.5.1	4282146
1.1.3.4.2	VPBC opy	1.1,8	ResetCurrency
1.1.3.4.3	YP8Link	1.1.8.1	NextCDS
1.1.3.4.4	vessind	1, 1, 8, 2	PointComp
1.1.3.4.5	YPEFree	1,1,8,3	Ruffrustidate
1.1.3.5	Selvps	テーブル し	
1.1.3.6	FindVID	本発明によれば、メディアが不確実であるか否	
		and the second of the second o	

本発明によれば、メディアが不確実であるか否か又はメディアが最初にアクセスされたか否かが呼び出される。本発明のポリュームマネージメント機能(ポリューム管理機能)はライン1、により表される。最初のプロセスは、ライン1.1 で示すように、どのポリュームがシステムに表されたかを決定することである。ライン1.1.1 のProbe-

Changeは、デバイスドライバにアクセスすべく呼 び出され、メディアの変化をデバイスドライバが 検出したか否かを決定する。メディアの変化が検 出されたときは、ライン1.1.2 においてRosetRedia が呼び出され、デバイスドライバがメディア に1/0 できることを知らせる。次に、ライン1.1. 3においてGentYPB が呼び出され、ポリュームパ ラメークブロックが創出される。このブロセスは、 LockVBufが呼び出されてオペレーティングシステ ムのカーネル内のバッファをクリアしかつ直列化 (serialize) するライン1.1.3.1 と共に開始する。 ライン1.1.3.2 においては、メディアブートセク タのデータが、オペレーティングシステムのバッ ファに読み取られる。システムはライン1.1.3.3 に続き、該ライン1.1.3.3 においては、BulldBPB が鮮び出されて(Invoked) 、ディスクドライバを 呼び出し(call)、ブートバラメータブロックを作 る。次に、FS_MOUNTがライン1.1.3.4 に呼び出さ れる。PS_Mount における最初のステップは、ラ イン1.1.3.4.1 の8*g_Set を呼び出す。ライン

Disklo

VPSFIND

Sag Set

VPECopy

WSPLink

CRC

1.1.3.7

1.1.3.8

1.1.3.9

1.1.3.10

1,1,3,11

1.1.3.12

1、1、3、4、1 は、898 のバッファを設定すべく呼び 出されるカーネルにおけるメモリマネージメント ユーティリティである。ライン1.1.3.4 において は、FSHouniVolume が呼び出されると、FSHouni-Volumeは、FSOsのリストを介して反復し、サクセ スに戻るかリストの終部 (エンド) に到達するま で、各PSB のFS_Mount 手順を呼び出す。ライン 1.1.3.4.2 においてタ\$B がサクセスに戻ると、ター PBCopyが呼び出されて、BPB のコピーに対するテ ンポラリバッファを創出する。次に、ラインにし 3.4.3 公的计名VPBLiek が呼び出され、VPB を予 ェーンにリンクさせ、孩チェーンにおける次の VPB を指すべくBPB を設定し、現在のVPB をリス トの開始に初期化する。VP8Find がライン1.1.3. 4.4 に呼び出されてVPBsのチェーンを試験し、ブ ロセス中のVPS と同じボリューム機関子を所有し ているVPS を見出す。複製VPS の識別子が見出さ れた場合には、ライン1.1.3.4.5 にVPBfree が呼 び出され、複数VPB がVPBsのリスト内に見出され た場合には、試験を受けてVPB が8P8 から自由に

なる、FSMoustVolume が完了すると、VPB に適当 なフィールドを設定するライン1.1.3.5 にSet\$P8 が呼び出される。ライン1.1.3、6 において、タindVIOが呼び出され、ボリューム識別子が見出され る。メディアのセククタにブートプロックが晃出 されない場合には、ライン1、1、3、7 にBisk10が呼 び出され、ボリュームのSPS が位置付けされる。 859 の85_Nount ルーチンがサクセスに戻らない 場合には、 (残留) FAT ファイルシステムのFS.... Mount 手順と論理的に等価のインラインコードが 呼び出される。ライン1、1、3、8 においては、CRC が呼び出されて、古いPAT ボリュームの最初のデ ィレクトリが検査合計(checksum)され、それらの ブートセクタの連続器等をもたないポリュームに ついてのユニークなポリューム連続番号が創出さ れる。次に、ライン1.1.3.5 ~ライン1.1.3.13に リストアップされた機能が呼び出されて、新しい ボリューム機関子が創出されかつボリューム機関 子バッファが空にされる。ライン1.1.2.14におい てはBullavalidate か呼び出されて、プロセス第 始以来メディアが変化している場合にはバッファ 内の全てのデータが無効にされる。その場合には、 ライン1.1.3.15にFlushBolが呼び出され、新しい メディアに対してバッファをフラッシュさせる。

ボリュームについて前から存在しているVP8 が 見出されない場合には、ライン1,1,4 のIncVP88efが呼び出されて、現在のVPB の基準カウンタが 増大(Increment) される。この基準カウンタは、 ここで問題にしているボリュームが、オペレーテ ィングシステムのカーネルに対して依然として関 放しているか否かを記録するのに使用される。ラ イント1.5 においては、DecVP6Ref が呼び出され。 前のVPB の基準カウンタが減少(decrement) され る。基準カウンタがぜ口に減少された場合には、 vappree がライン1.1.5.1 に呼び出され、VPB が 空にされる。ライン1.1.6 にはBesetCorrency が 呼び出され、現在のディレクトリ構成における位 置データが無効であるとしてマークする。Nextic-OS (ライン1.1.8,1)及びPointComp(ライン1.1.6. 2)は、現在のディレクトリ構成(CBSs)を列挙する

のに用いられる内部ルーチンである。ライン1.1。 5.3 において8vilevalidate が呼び出され。ファ イルシステムのバッファブールから、〈現在は陳 寫化している〉VP8 基準が除去される。

上記のように、VPBは、コンピュータシステムに使用されている特定のボリュームについての情報を記憶するシステムにより用いられる。ボリエームは、ブロック装置におけるメディアとして構成され、メディアに関する情報は、このボリュームを他の全てのボリュームから区別する。

VPBsは、BMP としてセグメントに維持される。 従って、システムは記録が使用されているトラッ クのみを必要とし、かつ窓のリストが管理される。

新しいポリュームに出合う度毎に、すなわち、 ポリュームのVPB 作製(VPB bnilt) がシステムに 既存のいかなるVPBsとも一致しない度毎に、新し い人口(new eniry) が、BMP 普理されたセグメン ト内で割当てられ、メディアからの等価データで 売譲される。システムがVPB で完了する度毎に、 すなわち、システムのRefCountがゼロになる度毎 に、880 管理されたセグメントの入口が空になり、 88P は、この空になった記憶機構(storage) を再 使用のためにトラックする。テーブル1の機能に より用いられた構成を以下に説明する。

yps は、次の3つの部分に分割される。

1.カーネルのブライベート部分。この部分は、 情報をカーネルに維持するのに使用され、¥P8(例 えば、基準カウント)の管理を必要とする。これ は、カーネルにとってのブライベートなものであ り、FSBsは決してこのブライベート部分にアクセ スしないしかつこれを変更することがないことを 意味している。

2.ファイルシステムの独立部分。この部分は、全てのファイルシステムにより使用されかつ特定のあらゆるファイルシステムから独立している。この部分は、或るファイルシステム(file system、「FS」)の要求に応じて、設置可能なファイルシステムに導かれる。

3、VPS を用いているファイルシステムに特有 の部分。この部分は、必要に応じてファイルシス

リント 可能な 10

テムを使用できる「作業領域」として設定される。 この部分は、成るPSの要求に応じてIPS に導かれ る。VPB のレイアクトは第7回に示されている。

次の構成は、ファイルシステムのVP8 とは独立 した部分を明らかにするものである。この構成は、 ファイルシステムの形式の細菌に係わりなく、あ らゆるファイルシステムにより使用できるもので ある。

vobisi STRUC

vol_10 00 7:77420324712=

~ 210

voi_papa BB マ:ドライブポリューム内概

vol…cbSector DW ?:バイトの物理セクタのサ

1 7

vpi_totaec BB フェメディアのセクタの全数

vpl_trksec OH ?:メディアのトラック当り

のセクタ

volumesed SW 7:装置のヘッドの数

vel text on venterties oup (?) ; 2 - Y

用のブ

vpbfsi ENOS

下記の構成は、VPS のファイルシステムの従属 部分を定めるものである。この構成は、適合する と考えられるファイルシステムにより使用される。 vobfad STRUC

vpd work DB VOBMORKANEASIZE DUP (?)
vpbfsd ENGS

下記の構成は、ボリュームバラメータブロック (8PB)の構成を定めるものである。

vpb STRUC

全てのファイルシステムについてカーネルにより 使用されるフィールド (欄)

*pd_filak OM ? :前方リンクのハンドル

vpf_blick DB ? i独方リンクのハンドル

vpb_10zector 98 ?:18のセクタ数

vpb_ref_count SW 7;VPB を指す目的のカウン

1

vob_search__ 8M ?:VPB を指すサーチのカウ

tount 2 h

vpb_first_ac BB ?;これは、メディアを強制

cess すべく-1に初期化され

Š

uph_signature ON 7;VPB の有効性を明配する

447

*pb_flags 08 9:75%

vpb_FSC DB ?:ファイルシステム制御ブ

ロック(FSC) を指すボイ

14

下記のフォールド (欄) は、ファイルシステム 健嘱作業に使用される。

vph_fed OB SIZE vphfed BUP (7)

下記のフィールドは、ファイルシステム独立作 単に使用される。

vpk fai BB SIZE vpbfai DUP (7)

vab 888S

下記の構成は、FS<u>H_</u>CETYOLFARK(これは、VPR ハンドルからVPR データを得るのに使用される) により使用される。

ENTRY push word hVPB (1 word)

push dword ptr to file

system ind. (2 word)

push dword ptr to file

system dep. (2 word)

call FSHGETVOLPARM

EXIT (ax) = return code (9/2-13-4)

0-success

下記の構成は、FSM BOVOLIO (これは、ボリュームベース形セクタ配列転送 (volume-based sector-oriented transfers)に使用される) により使用される。

ENTRY pash word Operation (1 word)

push word hopk (1 word)

push dword ptr is user

transfer area (2 word)

pash dword pir to sector

count (2 word)

push dword starting sector

(BS:SI) AVPSUL, FELVIAT

(AX) = 1/0 packet status:作業は

びこの中の全てのポリュームは

898 へのハンドルであ

がエラーを創出した。

AX, SX, DX, DI, BS. Flags

ュームで充壌される。

下記の呼び出しは、ボリェーム管理の内部コン

GentVFB は、特定のドライブの内部VFB の決定

に使用される。戻された全てのエラーは、ユーザ

laputs: datal 関心をもつDPB への点。これ及

コックされる。

ă., Carry set = >作業 (オペレーション)

zero clear =>作業は失敗した。

Outpute: Carry clear = >axは、ドライブを行う

Carry Set =>

失敗した。

ボーネントインターフェースに使用される。

តែបស់ស៊ីខាវ

(2 word)

call FSHOOVOLIO

EXIT (ax) = return code

Sections.

下記の構成は、FSE_DUPLICATEVPS (これは、複 製(古い)VPBへのVPS データを得るのに使用され る) により使用される。

ENTRY push word hypn

(l word)

USES

に送られる。

push dword our to file

system ind.

(2 word)

push dword atr to file

system des.

(2 word)

call FSNGRTVOLPARM

SXIT (ax) = return code

O-sucress

RedetermineMediaは、下紀に示すような特殊な 組の人口バラメータ(entry parameters)を有して 41 3 a

ENTRY (05:31) dob ~ Ø & (point)

EXIT

Carry clear = >

- 全てのレジスタは、BPを除く全てを変更した。

- FSMountVolume は、IES ドライバが、関心をも つポリュームを認識しているか否かを決定すべく チェックする。

FSHountVolume は、各FSドライバのFS..... Hount 入口点を呼び出すPSD チェーンを通してループし、 IFS が、関心をもつポリュームを認識しているか 否かを決定する。このルーブは、最初のIFS がボ リュームを認識するとき、又はシステム内に設置 されたFSドライバの番号のループカウンタが Bに

Toputa: ds:bx pVPBBufブートセクタへの点。

di xyyyolocalVP8のオフセット Outputs: di ~ IFS がボリュームを認識した場合

のFSC へのオフセット

di m ml 。 いかなるIFS ドライバもボ リュームを認識しなかった場合。

AX WYDD XYYN

3.3

zero set w >発生した不確実メディア を入れ子にした(nested)。

全てのレジスクは、変更することができる。

Buildd8P8 は、古いディスク(すなわち、認識 されたブートセクタを有していないディスク)展 の正しい80% を創出すべく呼び出される。新しい。 ディスクは、ブートセクタ内に、88088(既知)で 正しいすなわち正当なBPB(V&LID BP8)を有してい る。デバイスドライバへのバッファは、Boild8F8 呼び出しの一部である。

Inputs: daysi 関心をもついるへの点。

pVP88sf は、ロックされる。

Outputs; carry clear = >

ds:si BPBを指す。

carry set = >

(AX) =装置からの状態ワード(status

zero set =>不確実メディアを入れ子 にした。

zero reset =>作業は失敗した。

減少するときに、終端する。

変更されたレジスタ: ax, bp, bx, di, es, zi,

VPBFree は、リンクリストからVP8 を除去して、 セグメントからそのブロックを空にする。

ENTRY (SP) = VPB ~ On > FN

BXII リンクされておらずかつ空にされたVPB

USES bx. bp. cx. di. ds. es

VPBLink は、リストの開始時に新しいVPB をインサートし、新しいVPB 及び古い最初のVPB の部 方及び後方のリンクフィールドを調節する。

ENTRY ES: BI - MILL STYPE

EXIT TX FET Y 2 SAR VPB

uses Ds. S1

VPBFind は、内部リストを走査して、入力VPB と同じポリュームIDをもつVPB を探す。

ENTRY DS:SI - 入力VPB のボリュームIDを指す ポインタ

gait ax-bypa、晃出された場合

ax=0、見出されない場合

USES AN, BN, CN, DI, DS, ES

VPBCopy は、ローカル領域からBMP 管理領域に VPB をコピーし、かつ正当であるとしてVPB そス

れを含むことがある作業を実行する、FSO は、ポリュームラベル又は連続番号が変更された場合には、いつでもポリュームバラメータブロック(VPB)を更新する。

FSB が3/8 リクエストをFSヘルバルーチンに導 くと、デバイスドライバは、32ピットのポリエ 一ム連続番号及びボリュームラベルを(VPB を介 して) 導く、ボリュームに関して1/9 が実行され るとき、オペレーティングシステムのカーネルはく リクエストされたボリュームの連続番号を、装置 を維持している現在のポリュームの連続番号と比 較する。これは、ドライブにマウントされたポリ ュームのドライババラメータブロック(808) の808 をチェックすることにより行われるインストレー ジ試験(in-storage test) であり、いかなる1/0 も必要とされない。比較の結果、等しくない場合 には、オペレーティングシステムのカーネルはい クリチカルエラーハンドラに信号を発生して、特 定の連続番号及びラベルをもつポリュームをユー ザがインサートすることを促進させる。

タングする。

BRIRY SI = 29,00 Local VPBO * 7 to 1

85:01-># UNYP8

exit at

uses ax, cx, os, st

悪いポリュームがマウントされたときにこれを 検出してオペレータに正しい処理をとるように知 らせること、すなわちポリュームマネージメント (ボリューム管理) は、オペレーティングシステ ムのカーネル及び適当なデバイスドライバを介し て直接行われる。本発明の原理によれば、各ファ イルシステムドライバ (FSD)は、ポリュームラベルと、ファイルシステムに使用される各ボリュームについての32ピットのボリューム連級番号と を創出するようになっている。これらは、ポリュームがフォーマット化されるときに、論理的かか ましい。この情報を記憶するのに、特別のフォーマットが必要になることはない。オペレーティン グシステムのカーネルは、FSD を呼び出して、こ

メディアの変化が検出されると、アプリケーションプログラムのインターフェース(8PI)の機能 呼び出しの代わりに、ドライブがアクセスし、本 発明によれば、そのボリュームに対するマネージ ング1/0 に応答できるファイルシステムドライバ (FSB) が決定される。次に、本発明は、ボリュー ムバラメータブロック(VPB) を割当て、設置され たFSO2をボーリング(poli)する。FSB は、これが メディアを認識していることを表示する。FSBsは 上記のようにしてボーリングされる。

PAT のFSD は、PSDsのリストの整後にあり、他のFSD の認識が生じない場合には、全てのメディアを認識することにより、省略時FSD (default FSD)として作用する。

本発明の原理によれば、ファイルシステムドラ イバには、次の2つのクラス。すなわち、

1、ローカルすなわち遠隔(仮想ディスク)装置に対して1/8 を行うブロックデバイスドライバを用いている838 (これは、ローカルファイルシステムと呼ばれている)と、

 プロックデバイスドライバを用いることなくして達職システムにアクセスするPSD (これは、 遠隔ファイルシステムと呼ばれている)とがある。

ドライブレター(ドライブ文字)と適關ファイルシステムとの間の連結はプログラムされたインターフェースを介して行われる。システムのネームスペース(例えば、ドライブ)の目的とFSD との間の結合を生じさせるには、DosFSAtiach システムの呼出しが用いられる。

疑似文字装置 (pseudo-character device) た選 関ファイルシステムとの間の連結も、BosPsAttach インターフェースを介して行われる。BosPsAttach インターフェースは、BosPsAttach 呼出し及び BosSPsAttach呼出して構成されており、これらは、 Appendix Iにおいて詳細に説明されている。

ローカルボリュームが最初に参照されるとき、本発明では、FSO チェーンの各ローカルFSD を連続的に凝ねて(ask) 、各FSO のFS_ROBST 入口点への呼出しを介してメディアを受け入れるようになっている。どのFSO もメディアを受け入れない

が比較される。試験が成功した場合には、ポリュームにFSB がアクセスされる。これに対し、試験が失敗した場合には、オペレーティングンステムがクリチカルエラーハンドラに信号を伝達し、ユーザがポリュームを矯正することを促進させる。

メディアと\$P8 との間の連結は、ポリュームの全ての開放ファイルが関じられるまでセーブされ、サーチリファレンス及びキャッシュバッファリファレンスが除去される。ボリュームの変化によってのみ、次のアクセス時にメディアの再決定がなされる。

ブート可能で論理的に区分 (partition)された メディアに関するオペレーティングシステムの区 分へのアクセスは、08/2オペレーティングシステムに利用できる機能セット (機能組) のようなフルオペレーティングシステムの機能セットを介して行われる。ディスクの区分化の設計(disk partitioning design) についての詳細な説明が、前述の08/2プログラフ用参考書においてなされている。 場合には、メディアは、省略時ファイルシステム に割り当てられる。FORMATでは認識されないメデ ィアにアクセスするため行われる他の全ての試み がなされて、「無効(正しくない)メディアフォ ーマット(INVALIO HEBIA FORMAT)」のエラーノ ッセージが出される。

ひとたびボリュームが認識されると、ドライブと、PSD と、ボリュームの連続番号と、ボリュームの連続番号と、ボリュームの連続番号と、ボリュームのが発情される。ボリュームの連続番号及びラベルは、ボリュームのバラメータブロック(VPB) に記憶される。VPB は、開放ファイル(I/6に基づくファイルハンドル)、サーチ及びバッファリファレンス(buffer references)についてのオペレーティングシステムにより維持される。

除去されたボリュームに対する連続リクエストは、FS_MOUNTを呼び出すことによりボリュームについての設置されたFSOsのボーリングを要する。 認識しているFSO により戻されたVPB 及び既存の VPB のボリューム連続番号及びボリュームラベル

本発明によれば、ネットワークを介してオペレーティングシステムと通信する遠隔装置を識別するためのBosQFsAitach機能が提供される。BosQF-satiachの目的は、取り付け(sitach)られた遠隔ファイルシステム、ローカルファイルシステム、文字装置(characian device)、又は、ローカルPSB又は遠隔FSD に取り付けられた疑似装置ネームに関する機器を概ねることである。

Box & Prattiach を呼び出すシーケンスは次の通りである。

EXTURN Desufsattach: FAR

PUSH ASCIIZ DeviceName :デバイスネーム又は

PUSS WORD Ordinal :ネームリストの人口

PHSH HONO FSMinfolevel:要求される、取り付

けられたFSO データ

の序数(ordinal)

の形式

PBSH OTHER DataBuller (漢されたデータパッ

7 7

PUSH WORD DataBufferton;バッファ長さ PUSH DWORD O ;リザーブド(Oでなく てはならない)

CALL Bos@FsAttach

ここで、BeviceNameは、コロン(1)を付した駆動文字(drive fetter)を指すか、或いは文字又は疑似文字装置ネームを指し、PS&fefotevelの機つかのバリューは無視する。BeviceNameが文字又は疑似文字であるときには、BeviceNameは、コロンを付した駆動文字の形態を有するASCI2ストリングである。BeviceNameが、文字又は疑似文字の装置ネームであるときには、そのフォーマットは、サブグイレクトリ呼出しのファイルネームのフォーマットにおけるASCIIストリングのファーマットであり、DEV、のように示すのが好ましい。

序数(ordinal) は、文字装置、疑似文字装置又はドライブの組のリストのインデックスである。 序数は常に1からスタートする。リストの1つの項目の序数位置は重要でない。序数は、リスト全体を厳格にステップするのに使用される。序数か

unsigned short chiase;
unsigned char salase [];
unsigned short chfSDNsse;
unsigned char safSDNsse [];
unsigned short chfSAData;
unsigned char refSAData [];

1

Nyse 項目の形式

l = 常數文字裝置(Resident character device)

2 - 疑似文字装置

3 = ローカルドライブ

4 ≈PSD に取り付けられた遠隔ドライブ

chiane 項目ネームの長さ、ゼロ(null)は数

度ない

sikame 項目オーム、ASCIIZストリング

cbf58Maxe F30 ネームの長さ、ゼロは数えない

azFSOName ASCIIZストリングに取り付けられた

788 項目のネーム

cofsabata FSB により戻されたFSB アタッチデ

ータの最き

ら項目へのマッピングは揮発性であり、BosSFALtachに対する或る呼出しから次の呼出しまで変化 することができる。

FSAInfolevelは、要求される情報のレベルであり、BateBufferのデータがどの項目に関するものであるかを決定する。

レベル 0 × 0001は、BeviceNameにより名付けられた特定のドライブ又は装置ネームのデータを 質す、序数のフィールド(攤)は無視する。

レベル 0 × 0002は、序数により選択された文字 又は疑似文字装置のリストの人口のデータを戻す。 BeviceNamesのフィールドは無視する。

レベル 0 × 0003は、序数によって選択されたド ライブのリストの入口のデータを戻す。Bevicusame のフィールドは無視する。

SalaBufferは戻り情報バッファ(retorn information buffer)であり、次のフォーマット内にある。

struct (

unsigned short iType;

rgPSABata PSB により戻されたPSB アタッチデータ

a2FSOName は、FSO により移出(エクスポート) されたFSO ネームであり、このネームは、必ずし もブートセクタのFSO ネームと同じネームである 必要はない。

ローカル文字装置(iType = 1) については、 cbFSBName = 0 であり、saFSBName は、ゼロで終 わるバイトのみを含んでいて、cbFSABata = 0 で ある。

ローカルドライブ (iType = 3) については、
szFSBHane は、野出しの時点でドライブに取り付けられたFSD のネームを含んでいる。この情報は
ダイナミックに変化する。ドライブがオペレーティングシステムのカーネルの常駐ファイルシステムに取り付けられている場合には、szFSBHane に
"FAT" 又は"BHKSBUSK" が含まれるであろう。常駐ファイルは、マウントを抱むFSDs以外の任意のディスクに取り付けられるので、認識可能なファイルシステムを含んでいないディスクを設けること

ができるが、常駐ファイルシステムに取り付ける こともできる。この場合、差異を検出することが でき、この情報は、破壊されないデータのプログ ラムが、適正に認識されなかったディスクに存在 することを助ける。

BataBufferton は、関リバッファのバイト最き である。関り時においては、この長さは、PSD に よりBataBufferに関されたデータの長さである。

Returns: IP ERROR (AX not = 0)

AX - Error Code:

ERROR_INVALIO_DRIVE - 特定のドラ イブは無効 である (正 しくない)。

ERROR__8UFFER__OVERFLON- 特定のバ ッファは、 戻されたデ ータにとっ て非常に短

3 Y ...

のファイルシステムをカーネルに取り付けたか否か (他のPSOsはディスクを取り付けていないからである)の決定を行うのに使用することができる。

全てのFSDsへのエラー信号についてのエラーコードの組は、0×EESO-0×EEFFである。必要に応じて他のエラーを付加できるけれども、次のエラーが定められている。

ERROR VOLUME_NOT_NOUNTED = 0×8800-PSDは、 ポリュームを認識しなかった。

各FSB により定められるエラーコードの組は、 O×SPSG-O×FEFFである。

ディスクメディア及びファイルシステムのレイ アウトは、次の構成により説明される。ファイル システムに与えられるデータは、ブロック装置に 取り付けられたデバイスドライバにより与えられ るファイルシステムサポートのレベルに基づいて いる、これらの構成は、ローカルファイルシステ ムに対してのみ等犠牲を有している。

/*file system independent-volume params*/
struct upfsi (

SRSOS_NO_NORS_ITEMS-特定の序数

は、リスト

にない項目

に関するも

のである。

ERROR INVALIGUEVEL- IEL CALL

情報レベル

(invalid

isfo level)

全てのプロック装置及び全ての文字及び疑似文字装置についての情報は、Bos8Fshttachにより戻される。この呼出しにより戻される情報は、深発性が強い。

関された情報は、これらが関されるときまでに 既に変化されていることを呼出しプログラムが気 付くことができるようにするのが好ましい。カー ネルの常駐ファイルシステムに取り付けられたデ ィスクに戻された情報は、ディスクがファイルシ ステムを備えたものであることをカーネルが明確 に認識しているか否かの決定、又はカーネルがそ

unalgood long voi_vid; /* 32E > F#

9 2 - 218*/

unsigned long spi_bBEV: /*デバイスドラ

イバベのハンド

14.

unsigned short vol bsize: /* Mit D t 2

クサイズ*/

unsigned long vpi_totsec: /*セクタの全数*/

unsigned short vpi_trksec; /*セクタの全数*/ unsigned short vpi_nhead; /*セクタの全数*/

chert voltext [12]; /* ascliz#9 x

~ L 2 ~ L °/

1 ; /**pfsi*/

/* [ile system dependent-volume params*/
struct vpfsd (

char vpd_gark [36]; /*作業領域*/

); /* vpfzd*/

上記のように、PS_MOUNT機能は、ボリュームを マウントしかつアンマウントすべく呼び出され、 その目的は、PSD がファイルシステムのフォーマ ットを認識しているか否かを決定すべくポリュームを試験することにある。PS-MOUSTを呼び出すシーケンスは下記の適りである。

int far pascal FS_NOUNT(flag, pupisi, pupish, about)

unsigned short flag; struct upfsi far* pupfsi; struct upfad far* pupfad; unsigned short byPB;

char far" phoot:

ここで、フラグは、要求された作業を示す。

flag = 0は、ボリュームをマウント又は受け入れるのにFSD が要求されることを示す。

fias - 1 は、特定のポリュームが除去されたことを示す。

flag = 2は、ボリュームがそのドライバから終 去されるときに当該ボリュームに割り当てられる 全ての内部情報及び当該ボリュームが除去された ことの最終のカーネル管理リファレンスを開放す るのにPSB が要求されることを示す。

指すポインタ、このポインタは、フラグ == 0の ときにのみ正当である。ポインタのバッファは、 IMUST NOT 88 MODIFIED (変更してはならない)」 を指示する。ポインタは常に正当であり、フラグ == 0 である場合にはその正当であることを確認 する必要はない。緩取りエラーが生じた場合には、 パッファはゼロを包含する。

FSD は、もたらされたボリュームを試験し、ボリュームがファイルシステムを認識しているか否かを決定する。ボリュームがファイルシステムを認識している場合には、vpfxi 及びvpfsd の適当な部分を充職した後に、ゼロに戻る、vpi_vid 及びvpi_textのフィールドは、FSD により充壌される。FSD がオペレーティングシステムのフォーマットブートセクタを有している場合には、PSD は、メディアをラベルからssclisフォームに変換する。vpi_hOsvのフィールドは、オペレーティングシステムにより充壌される。ボリュームが認識されていない場合には、ドライバは非ゼロ(non-zero)に戻る。

他の全てのバリューはリザーブされる。FSB に 選かれるバリューは正しいであろう。

pxp(xi-YPB のファイルシステム独立部分を指すポインタ、メディアが、オペレーティングシステムの認識可能なブートセクタを収容している場合には、ファイルされたvpi_vid は、ボリュームについての32ビット識別子を収容する。メディアがそのようなブートセクタを収容しない場合には、FSD は、メディアに対してユニークなラベルを創出して、該ラベルをファイルされたvpi_vid に配置する。

pvpfsd-9F8 のファイルシステム従属部分を指すポインタ、FSB は、必要に応じ情報をこの領域に記憶させることができる。

8978-ボリュームへのハンドル。

phont ーメディアから読み取られたセクタ 0 を

voi_text及びvoi_vid は、これらのバリューが 変化する度毎に、FSO により更新される。

vpfsd の内容は次の適りである。

FLAG = 0

FSO はPSO FINDOUPHYPS を発行して、複数VPS が存在しているか否かを決定する。複数VPS が存在している場合には、新しいVPS のfs提高領域が正しくなく、FSO がFS_NOUNT の呼出しから戻った後に、新しいVPS がアンマウントされる。PSO は、古いVPS のfs提高領域を更新する。

複製ypg が存在しない場合には、FSD は、fs従 医領域を初期化する。

FLAG - 1

VPS のfs従属部分は、PSD が機律にこの従属部分を変更したものと同じである。

81.80 = 2

yps のfs従属部分は、FSB が最後にこの従属部分を変更したものと同じである。

メディア認識プロセスの後、F5<u>IL</u>GETVOLP&FMF 出しを用いて、ボリュームバラメータを試験する ことができる。ポリュームバラメータは、メディア認識プロセスの後に変更すべきではない。

マウントリクエストの間、FSB は、FSM_BOVOL-10を用いることによりメディアの他のセクタを試験し、1/0 を行うことができる。不確実メディアの関りが検出される場合には、FSD は「cleans ap (クリナップ)」の状態になってENROM_BNCEN-TAIN_HEDIAを関し、新しくインサートされたメディアに関してポリュームマウント論理が再スタートできるようにする。FSB は、付加1/0 に使用できるバッファを設ける。

オペレーティングシステムのカーネルは、上記 レフカウント(retcount)のカウンタを介してYPB を管理する。全てのボリューム特定目的(volumespecific objects) は、適当なボリユームハンド ルでラベリングされ、YPB に対する基準を示す。 ボリュームに対する全てのカーネル基準が消滅す ると、フラグー2でPS_MOUST が呼び出され、デ ィスマウントリグエストを表示する。

ボリュームがそのドライブから踏去されたこと

合(例えば、PSB がCB-RONのみをサポートする場合)には、作業は失敗するであろう。

精どのコンピュータンステムのハードウェアは、メディアのカーネルメディア形除去(Kernel-wediated removal of wedia)ができないので、ポリュームがどのドライブにも存在しないときにアンマウントリクエストが発行されることは確実である。FSH_00V0L10 は、特定のボリュームに対する1/0を実行する。PSH_00V0L10 は、リクエストがケットをフォーマット化し、デバイスドライバを呼出し、かつPSO に戻る前に、あらゆるエラーをハードエラーダエモンに報告する。ハードエラーダエモンに報告する。ハードエラーダエモンに報告する。ハードエラーダエモンに報告する。ハードエラーダエモンにより表示された全ての再試行(retries)又は200588808により表示されたアクションは、PSH_00V0L10 への呼出しを行っている間に行われる。

次に、FSM_DOVOLIO に対する呼出しフォーマットについて説明する。

int far pascal PSB 20VOLIO (operation, bVPB.

をカーネルが検出し、かつ、依然としてポリュームに対する未解決の基準(outstanding references)が存在すると、FS__MOURT はフラグ・1で呼び出され、FSO が、ボリュームについてのクリーンなデータ(又は、他の再生可能なデータ)を記憶できるようにする。ダーティで再生不可能なデータは、該データがドライブ内にリマウントされるときに、ポリュームに審き込むことができるように保持される。本発明の目的を得る上で、クリーンデータは変化されないデータである。

PData. poSec, 1Sec)
unsigned short operation:
unsigned short hVPB;
char far* pData;
unsigned short far* poSec;
unsigned iong iSec;

ここで、オペレーションビットマスク(speration bit mask) は、実行すべきread/read-bypass/write/mr

Bit 0×0001 offは、数取りを表示する。

Bit 0×0001 on は、数込みを表示する。

Bit 0 × 0002 offは、no-bypass を表示する。

Bit 0×0902 on は、casche bypass を表示す

Nit 0×0004 offは、no verify after write 作業を表示する。

8jt 0×0004 on は、verify after-writeを表示する。

8it 0 × 0008 offは、ハードエラーダエモンに 個号を送られたエラーを要示する。

Bit 9×0008 on は、ハードエラーが直接関されることを表示する。

Bit 0×0010 offは、1/0 が *write-through* でないことを裏示する。

8it 0×0010 on は、1/0 が *write through* であることを表示する。

8it 0×0020 offは、この1/0 に対するデータ を変更すべきことを表示する。

8it 0 × 0628 on は、この1/3 に対するデータ を変更すべきではないことを表示する。

リザーブされた他の全てのビットはゼロである。

「cache bypass (キャッシュバイバス)」と、「as cache (のーキャッシュ)」ビットとの間の 相違は、リクエストバケットの形式においてデバイスドライバが悪かれることである。「cache bypass」では、コマンドコード24、25叉は 26でパケットが得られ、「so cache」では、システムは、コマンドコード4、8叉は3に対する

ができる、FS_MOUNT呼出しの範囲内で呼び出され るとき、FSH DOVOLIO は、ボリュームの如何に係 わらず、ドライブのボリュームに適用される。し かしながら、PSD がPS_MOUNTの呼出しに関るまで は、ボリュームの認識が完了しないので、PSB は、 ERROR UNCERTAIN MEDIA WEET AUNCERUE. 特別な注意を払わなければならない。これは、ド ライブにおけるメディアを識別するのに、メディ アが不確実な試みを行ったことを表示する。また、 PSO が認識を試みたポリュームが除去されたこと を表示するようにしてもよい。この場合、FSB は、 FS_MOUNTの呼出し時に導かれたhVPBに取り付けら れたあらゆる質点を解放し、かつ88808_BRCERTA-18_MEDIA & FS_MOUNT呼出しに戻す。これにより、 マウントプロセスを再スタートさせるように、ボ リュームトラッキング論理が仕向けられる。

PSDsは、PSH_DOVOLIO2を呼び出して、I/O 作業 とは独立して、デバイスドライバの作業を制御す る。このルーチンは、IOC7L 作業に対するポリエ ーム智理をサポートしている。FSD に戻る前に、 バケットを拡大する。

hVPB 1/0の質測に対するボリュームハンドル。

pBsts ユーザの転送領域(transfer area) の 長いアドレス。

paSec 転送すべきセクタ数を指すポインタ。 戻り時には、これは首尾良く転送されたセクタ数 である。

iSec 転送の最初のセクタ数。

Returns -作業が失敗した場合には、0以外のエ ラーコード。

ESROR PROJECTION VIOLATION - 供給されたaddress/length は正しくない。

ESROE UNCERTAIN MEDIA - メディアが変更されているときには、デバイスドライバは信頼性をもって告げることはできない。このことは、FS_MOUNTのシテクスト内においてのみ生じる。

PSB BOVOLIO は、常時、FSB 内で使用すること

全てのエラーがハードエラーグエモンに報告される。ハードエラーグエモンにより表示されるすべてのあ試行又はBOSERRORにより表示されるアクシッンは、PSH_BOVOLIO2への呼出し内に行われる。int far pascal FSH_BOVOLIO2 (bBev. sfa, cat.func, pfara, cbPara, pBata, cbData)

unxigned long blov;

unsigned short sfa:

unsigned short cat;

unsigned short func;

char far* prara;

anglened short obpors:

char far* pDete:

unxigoed short abbata;

222

apev-vpB から得られたデバイスハンドル。 sts -psg_pevioctに呼出しを引き起こしたオ

ーブンインスタンス(spen instance) からのシステムファイルの数。このフィールドは、変化されない状態で、sfi selfsfn のフィールドから遅か

れるべきである。どのオープンインスタンスもこ の呼出しに一致しないときには、このフィールド は、0×FFFFに設定される。

cat -実行すべき10CTL のカテゴリ、

feac-10CTL のカテゴリ内での機能。

pPara ーバラメータ領域へのロングアドレス。

cbParwーバラメーク領域の長さ。

pBata ーデータ領域へのロングアドレス。

cbDataーデータ領域の長き。

Returns -エラーが検出されないときには0以外のエラーコード。

供給された機能が、ここに述べているシステムとの互換性をもたない場合には、EBROR_INVALID_FUNCTIONが呼び出される。メディアが不確実になるときにはいつでも、新しいVPB が割り当てられる(メディアが変更されていないことがもはや確かではないことを、デバイスドライバは認識している)。このVPB は、FS_MODNT呼出しが戻るまで、(メディアの両インサートにより)前に割り当てられたVPB と共に崩壊することはない。しかしな

がら、前のVPB は、メディア(このメディアは、 該メディアが除去されている間に書き込むことが できる)から更新されなくてはならない幾つかの キャッシュデータをもつことができる。古いVPB についてのキャッシュ情報(cached information) を更新するため、PSM_PINDDSPNVPB は、ボリュー ムの、この前に生じたことをPSD が見出すことを 可能にする。このボリュームについて別の古い VPB が存在しない場合には、新しく創出された VPB がアンマウントされる。

FSM_FINDBUPHVPR についての呼出しフォーマットは次の通りである。

int far pascal PSB FINUSUPHYPB (hYPB, phYPB) unsigned short hYPB;

unsigned short far phyPB:

2200

hyps-見出すべきボリュームに対するハンドル。 phyps ーマッチングボリュームのハンドルをど こに記憶させるかを指すポインク。

Returns ーマッチング998 が見出されないときに

はの以外のエラーコード。

ERROR_NO_ITEMS - マッチングhVP8は存在しない。
FSN_GETVOLPARNは、FSO が、VPB からファイル
システムの独立及び従属データを検索することを
可能にする。FSルータはVPB ハンドル内を通るの
で、ここのFSOsは、適合部分を指すポインタ内に
ハンドルをマッピングする。FSN_GETVOLPARNにつ
いての呼出しシーケンスは次の適りである。

void far pasce: PSB SETVOLPARN (AVPB, ppVPBfsi, ppVPBfsd)

unsigned short hVPB;

struct vpfsi far* far* ppVPBfsi; struct vpfsi far* far* ppVPBfsd; ZZZ;

BVPB-関心のあるポリュームハンドル。

ppVP8(siーファイルシステムの独立データを配 復きせるべくポインタが指す位置。

ppVPBfsd-ファイルシステムの従属データを配 憶させるべくポインタが論す位置。

Returns - W L.

FSE_Volumeのマッピングはダイナミックであり、かつPSO-DD連結は、FSB 及びDDの独立方法により、オペレーティングシステムのカーネルを介して行われるので、あらゆるPSD は、BBs がこのFSD からローディングされたポリュームを含むあらゆるポリュームにアクセスできる。ポリュームは、除去可能なメディアの特定のピース(片)、又は区分できる任意のメディアの任意の区分にマッピングするので、多数のFSDsが特定のハードディスク又は他のメディアにアクセスできるようにしてもよい。

ボリュームファイル作業は、2つのカテゴリ、
すなわち、ネームベース作業(named-based operations)及びハンドルベース作業(handle-based operations)に分けることができる。ネームベース作業は一般にユーザによって開始され、システム100がファイルについてネームド作業を行うことをユーザがシステム100に命令(instruct)する。ハンドルベース作業は、一般に、システムのバックグラウンド作業中に開始される。通常、

ハンドルベース作業は、ネームベース作業の後に 行われる。

第8回に示すように、コンピュータシステム 188かネームベース作業を行っていると来に、 ルーチン800が呼び出される。ネームド作業は、 文字ネームにより指示された作業である。すなわ ち、この作業は、ファイル又はディレクトリのネ ームにより特定化される。「Open lile "xxx"」 は、ネームペース作業の一例である。プロセス 802は、ネームをパーシング (parse)すべく呼 び出されて、3つの変数すなわち、PathWameType、 TCBThishVPB 及びTCBThisFSCを戻す。このプロセ ス802については、第3回に関連して詳細に説 明する。 {往、h はハンドルをいい、TCB は、 TCHThishVPE が現に関心をもっているVPS へのハ ンドルでありかつ1087bisFCBが関心のあるファイ ルシステムを指すポインタである。スレッド制御 プロック(thread control block)をいう。) 次に、 項目804は、プロセス802により戻された数 数PathType、TchthishVPB 及びTCBThisFCBに無ブ

正规明する.

次に、第9図を参照して、バーシングプロセス 802を説明する。呼出しがあると、項目902 は、現在のドライブ、現在のディレクトリ及びネ ーム自体に基づいて、関心のあるネームをカノニ カルフォーム (canonical form) に変形する。次 に、変数TCBTBISPSC、TCBThisVPB及Upathnesetype が下記のようにして決定される。デンジャン 906は、ユーザのネームが「\\」で始まって いるか否かを決定して、880 ネームが表示されて いるか否かを決定する。UNC ネームが表示されて いれば、劉樹が項目905に奪かれ、ここで、変 数oathives, feathishaps 及び feathisfes が初 期化され、ユーザのネームを適当な位置にルーチ ングする。UNC ネームが表示されていなければ、 デシジェン906は、関心をもつネームがカーネ ルにより維持された、袋器のネームリストに載っ たネームであるか否かを決定する。装置のネーム リストに載ったネームであるときには、デシジョ ン908は、それが疑似文字装置であるか否かを

く適当な機能に制御をルーチング(route) する。 UNC FSB が呼び出されるユニバーサルネーミング コンペンション(Universal Naming Convention) TUBC」)のグローバルネットワークを表示する 「\\」で経路(path)が始まるときには、項目 806を濁る制御が行われる。ローカル装置が表 示される場合には、制御が項目808に遅かれ、 カーネル内でリクエストが処理される。疑似装置 又は遠隔ファイルが表示されるときには、钢像は 項目810に導かれて、疑似装置又は遮隔ファイ ルが取り付けられた遺籍FSB に対するリクエスト をルーチングする。ネームドバイブ(named pipe) が検出された場合には、制御は項目812に悪か れて、カーネル無いでローカルネームドバイブコ - ドを呼び出す。ローカルファイルが表示される。 場合には、製御は項目814に導かれる。この項 图 8 1 4 は、項目 8 1 6 におけるFSHBOVOLIOを呼 び出すことによりポリュームへの誘取り及び搬込 みを行うFSD におけるFSD ワーカである。FSHDO-VOLIO については、第11個に関連して更に詳難

決定する。疑似文字装置であれば、項目 9 1 0 は 表示のように変数を設定する。疑似文字装置でないときは、制御は項目 9 1 2 に導かれ、表示のように変数を設定する。

デンジョン314は、ネームの初めにおける 「NoiseN」を探すことにより、そのネームが、 ネームドバイブであるか否かを決定する。そのネ ームがネームドバイブであるときは、項目916 は、表示のように変数を設定する。ネームドバイ プでないときは、デンジョン918は、そのネー ムがローカルドライブ又は遠隔ドライブのバスネ ーム (pathasse) を選示しているか否かを決定す る。選絡ドライブが表示されているときは、钢御 は項目320に懸かれ、ここでは、表示のように 数数PathType(TchThizhVPB 及びTCSThiafCH 宏設 遊する、遺職ドライブが表示されていないときは、 制御は項目922に導かれ、ここでは、どのボリ ュームから適当なデータを飲み取るかが呼び出さ れる。WhatVolumeが異ると、顕微は項目924に 薄かれ、ここでは、表示のように変数PathType。

TobThishypa 及びTCBThisPCHが設定される。

第10図を参照すると、アロセス1000が呼び出され、ハンドルペース作業が行われる。呼出し時に、項目1002はSPT 入口を検索する。SPT 人口及びハンドルは、両方共DosOpen により設定される。次に、項目1004において、FSC が指示するファイルシステムの適合FSB ワーカが呼び出される。次いで、項目1006は、項目1016を呼び出し、必要に応じて項目1016を呼び出すことにより、呼出し者すなわちコーラー(caller)によりリクエストされたあらゆる1/0 を実行する。

第11図には、FSB Bo Voi 10 が示されている。 項目1102において呼出しかなされると、hVPBが使用されて、ドライブ並びに関心のあるボリューム にどのようなボリュームがあるかを決定する。次 に、デシジャン1104は、ドライブにおけるボリューム アンジャン1104は、ドライブにおけるボリューム である、関心のあるボリュームであるときは、項目 1106が呼び出されて、デバイスドライバが呼び出

付録単は、ファイルシステムを用いることができるカーネルにより移出されるインターフェースのリストである。

付録では、本発明に従って構成された設置可能 なファイルシステムの一例の資源コードである。

付録Vは、付録NのFSD を作るべくOS/2により 用いられる定義ファイル(definitions file)のリストである。

以上、設置可能なファイルシステムにおいてダイナミックボリュームのトラッキングを行う方法及びシステムについて説明した。本発明の教示によれば、リンクされたリストにおいて編成される複数のファイルシステムドライバを、コンピュータシステムに設度に、省略特ファイルシステムドライバを設けることができる。コンピュータシステムに試更に、省略特ファイルシステムドライバを設けることができる。コンピュータシステムに新しいボリュームが与えられたとき、又はメディアが不確実になったときにはいつでも、コンピュータシステムは、ファイルシステムドライバがボリュームを受け入れるまで、自動的かつダイ

されかつ特定のパラメータで1/0 を実行する。改 にデンジョン1108は、作業性にメディアが不確実 に移行したか否かを決定する。不確実に移行して いないときには、プロセスは項目1114に戻る。デ ンジョン1108が、メディアは不確実ではないこと を決定するときには、制御は項目1112に導かれ、 ここでは、WhatVolumeが呼び出されてメディアが 確実なものにされる。次いで、制御はデンジョン 1104に戻る。ドライブにおけるボリュームが関心 のあるボリュームに一致しないときには、項目 1110が呼び出されて、HardError が呼び出され、 ドライブに正しいドライブを置くことをユーザに 知らせる。次に制御は上記項目1112に導かれる。

付録(Appendix) TからVは、設置可能なファイルシステムの質嫌の一例としてここに包含される。ここで、

付録 B は、本発明の数示に従ってサポートする ことが期待されるファイルシステムの移出インタ ーフェース (exported interfaces)のリストであ る。

ナミックにファイルシステムドライバを呼び出す ようになっている。どのファイルシステムドライ バもボリェームを受け入れない場合には、省略時 ファイルシステムがマウントされる。従って、当 業者には、本発明の他の用法及び変更が可能であ り、そのようなあらゆる用法及び変更は未発明の 範囲及び精神に含まれるものである。

4.図画の簡単な説明

第1 A 図は、本発明の原理に従って構成された コンピュータシステムのブロックダイアグラムで ある。

第18回は、第14回のコンピュータシステム の作業及びファイルシステムアーキテクチェアを 来すブロックダイアグラムである。

第2A図は、#S-BOSオペレーティングシステム のファイルシステム構成の詳細を示すプロックダ イアグラムである。

第28週は、本発明の設置可能なファイルシステムのファイルシステム構成の詳報を示すプロックダイアグラムである。

第3回は、第2日回のファイルシステムのより 詳細なブロックダイアグラムである。

第4回は、PATファイルシステムのディスクフォーマットを示すブロックダイアグラムである。

第5 A 図~第5 H 図は、本発明に使用できるように構成された設置可能なファイルシステムのディスクフォーマットを示すプロックダイアグラムである。

第6回は、本発明のマウントプロセスの全作業 の詳細を示すフローチャートである。

第7回は、本発明の設置可能なファイルシステ ムの構成を示すプロックダイアグラムである。

第8個は、本発明の原理に従がうネームベース 作業の実行の詳細を示すフローチャートである。

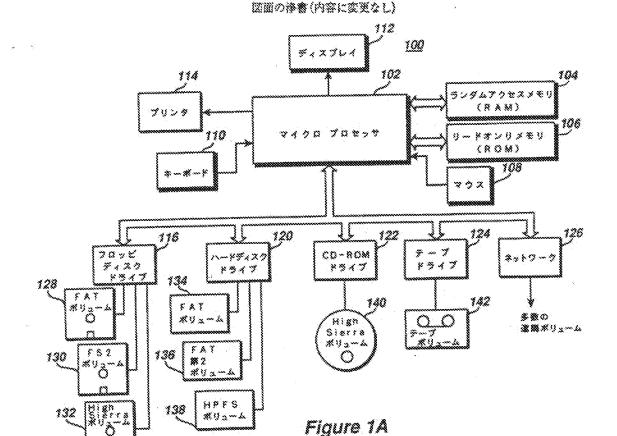
第9回は、ネームペース作業プロセスにより呼び出されるパーシングプロセスを示すフローチャートである。

第10回は、本発明の原理に従がうハンドルベ ース作業の実行を示すフローチャートである。

第11回は、第8回及び第10回に関連して数

関したプロセスにより呼び出されるFS<u>H_</u>BaValle プロセスを示すフローチャートである。

- 100 ~ コンピュータシステム、
- 102~マイクロプロセッサく
- 1 8 まっランダムアクセスメモリ、
- 106…リードオンリメモリ。
- 188 ... 773.
- 110~+~*~ *.
- 112…ディスプレイ、
- 114-7922
- 116…フロッビディスクドライブ、
- 1200ハードディスクドライブ、
- 1 2 2 ··· CD-ROM F 5 4 7 .
- 1210テープドライブ。
- 126~ネットワーク。



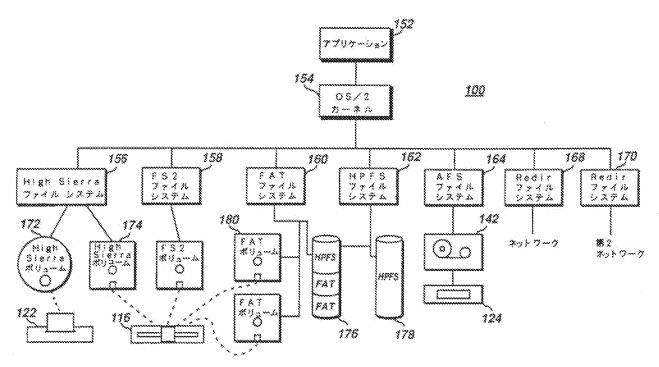
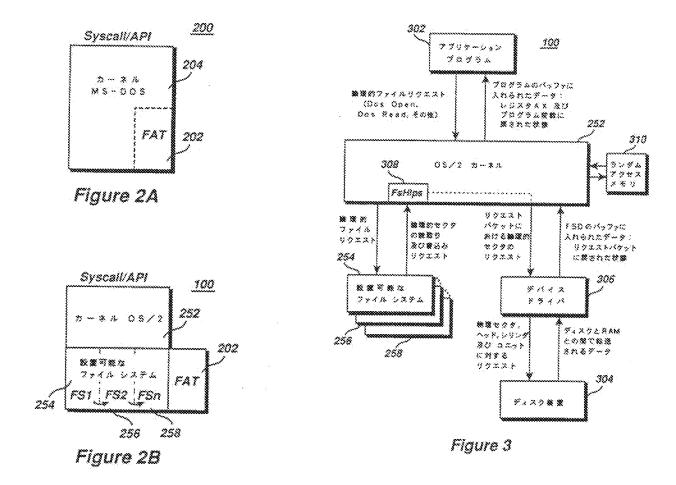
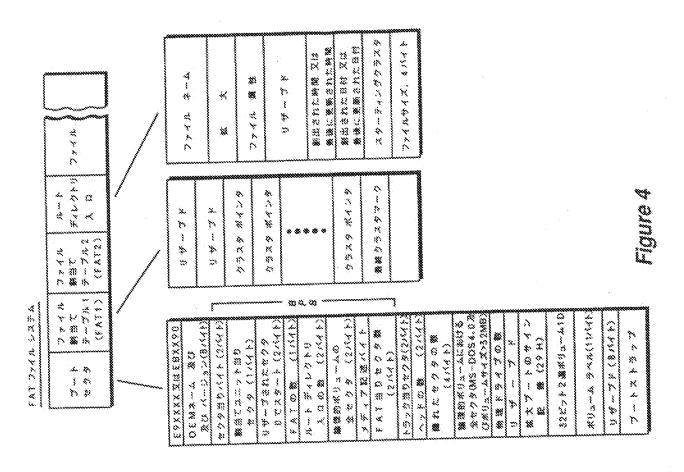


Figure 18





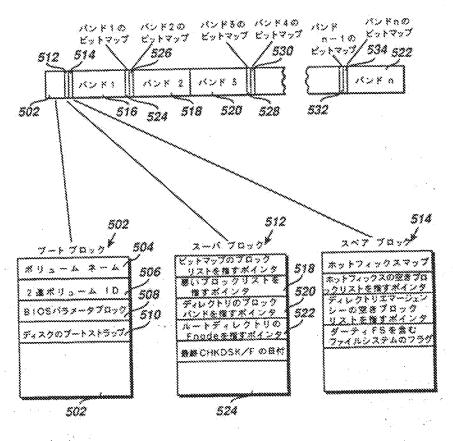


Figure 5A

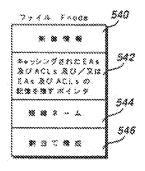


Figure 5B

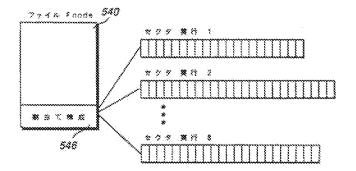


Figure 5C

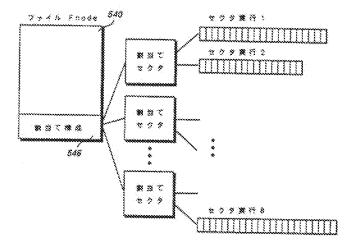
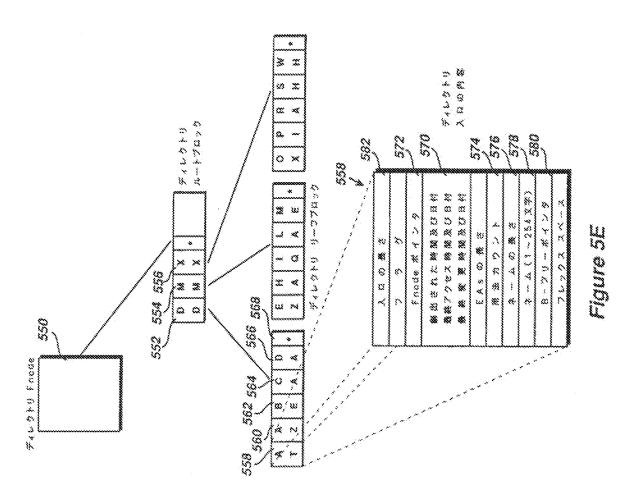
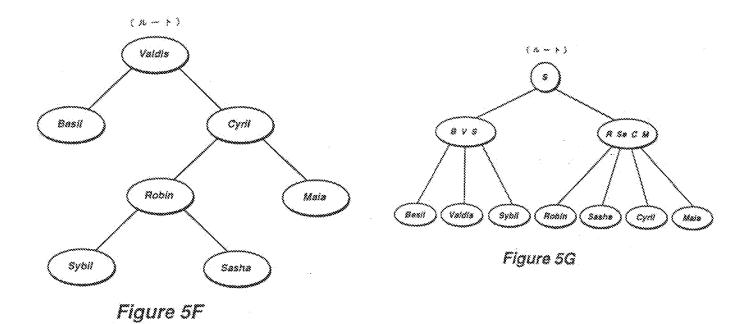
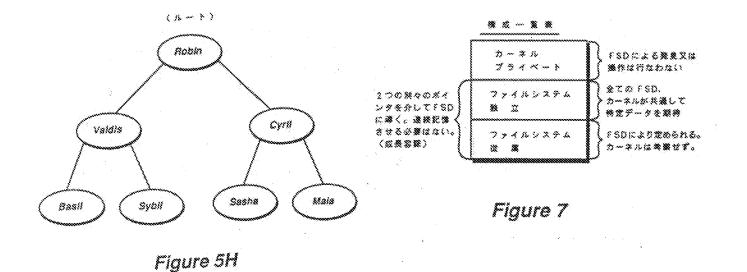


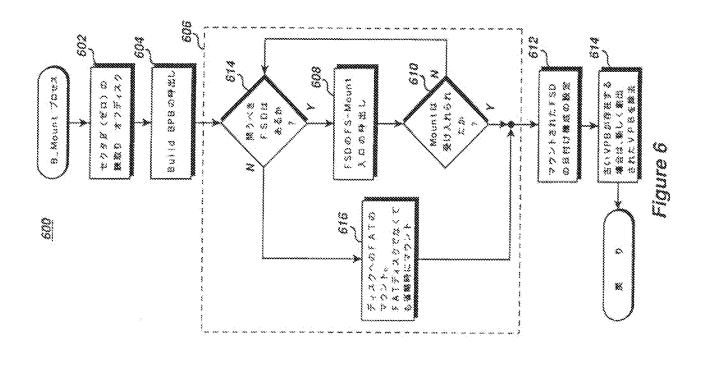
Figure 5D

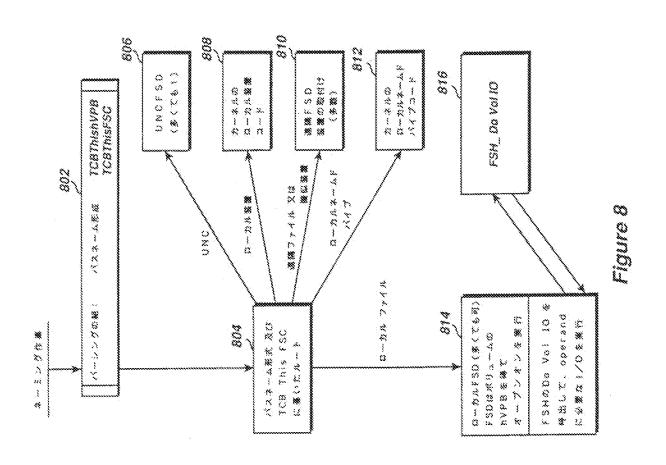


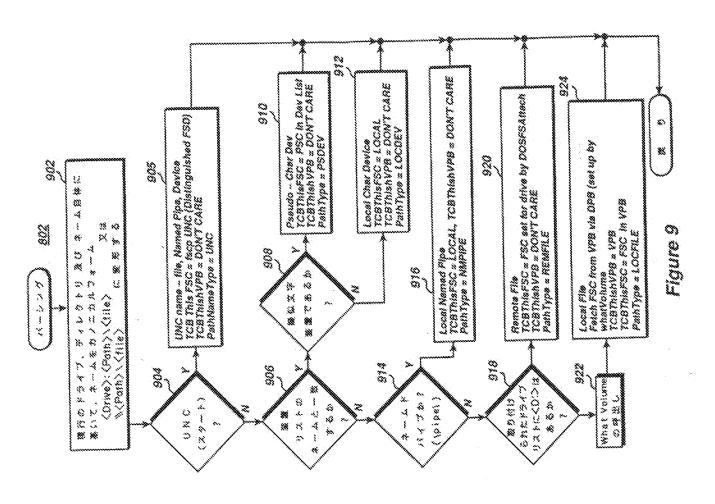
-231-

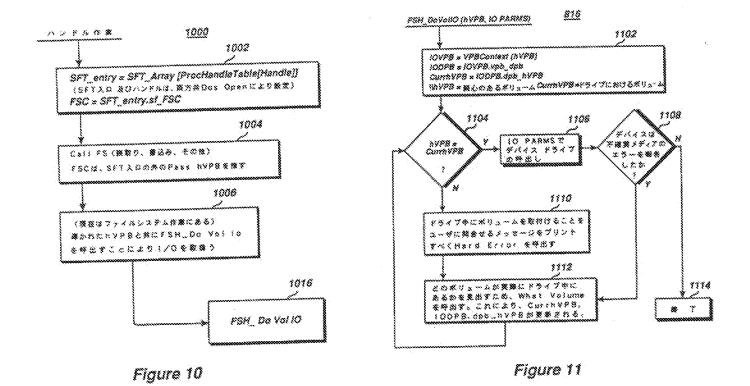












第1頁の続き

の発 明 者 マーク ジエイ ズビ アメリカ合衆国 ワシントン州 98072 ウッディンヴィ

> コースキ ルーノースイーストーウンハンドレツドアンドセプシティ

> > エイス プレイス 15817

砂発 明 者 ジエイムズ ジー・レ アメリカ合衆国 ワシントン州 98033 カークランド

> クウイン ノースイースト ワンパンドレッドアンドフオース スト

> > リート 11428

60条 明 者 ラジエン ジヤヤンチ アメリカ合衆圏 ワシントン州 98005 ベルヴィュー

> イラル シヤー ワンハンドレフドアンドトウエンティサード アベニュー

> > ノースイースト 517

手 粽 緒 正 實(方式)

2.12.26 华 月 平成 33

特許庁長官 摊 松 敏 毅 も事件の表示 平成2年特許翻案227905号

数数可能なファイルシステムにおいて ダイナミックボリュームトラッキング を行う方法及び数数 2.落納の名称

3. 矯正をする意

事件との関係。 出 額 太

マイクロソフト コーポレーション 8 8

毛代 寮 人

東京都王代田区丸の内3丁田3巻(号電路(代)211-8741

茂 念 (5995) 余建土 中 村

5. 横正命令の目付 平歧2年11月27日

長額正の対象 観客の特許出額人の概

代理機を証明する書面 全 38 16

2.輸出の内容 羽灰のと楽り巻 NSCHOLEGICE (PSCEECT)

2 12 27

....